

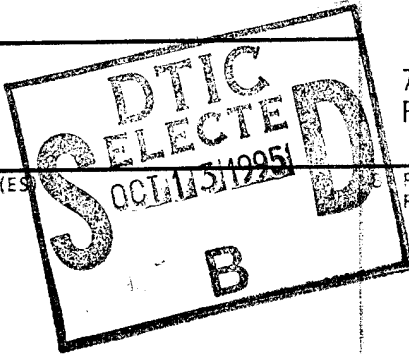
REPORT DOCUMENTATION PAGE

0599

18

Public reporting burden for this collection of information is estimated to average 1 hour per response, including gathering and maintaining the data needed, and completing and reviewing the collection of information, collection of information, including suggestions for reducing this burden, to Washington Headquarters, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

DATE SOURCE
ASPECT OF THE
41b JEFFERSON

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE		3. REPORT TYPE AND DATES COVERED FINAL/30 SEP 92 TO 29 MAY 95	
4. TITLE AND SUBTITLE NEURAL NETWORKS FOR INTERACTIVE IMAGE AND SIGNAL EXPLOITATION				5. FUNDING NUMBERS 7013/49 F49620-92-C-0072	
6. AUTHOR(S) JOHN PEARSON					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) DAVID SARNOFF RESEARCH CENTER PRINCETON, NJ					
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 110 DUNCAN AVE, SUTE B115 BOLLING AFB DC 20332-0001				10. SPONSORING / MONITORING AGENCY REPORT NUMBER F49620-92-C-0072	
11. SUPPLEMENTARY NOTES				19951011 150	
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Efficient reliable neural network design methods based on sound principles were developed. The main effort was concentrated on exploring the nature of error surfaces attempting to answer the question of how many minima actually exists in any given design. To this end the following experimental approach was taken: Define a scaleable model problem with a known solution Perform comprehensive, careful training experiments Refine training solutions to highly accurate minima Post process and count local minima Derive quantitative and qualitative measures that describe error surfaces. The Principal results show: How the number of minima varies as a function of the ration of training to network size the probability that training from a random start converges to a minimum How the number of minima increase with network complexity					
14. SUBJECT TERMS				15. NUMBER OF PAGES	
DTIC QUALITY INSPECTED B				16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
				20. LIMITATION OF ABSTRACT SAR(SAME AS REPORT)	

Neural Networks for Interactive Image and Signal Exploitation

Final Technical Report
October 1, 1992 - May 29, 1995

Contractor: David Sarnoff Research Center, Princeton, NJ
Principal Investigator: John Pearson, 609-734-2385

Contract Start: 9/30/92
Contract End: 5/29/95
Contract Amount: \$843,914

ARPA Program Manager: Barbara Yoon, (703) 696-2234
Program Code: 2D10

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 7103
Monitored by AFOSR under Contract F49620-92-C-0072

Neural Networks for Interactive Image and Signal Exploitation

Final Technical Report
October 1, 1992 - May 29, 1995

Table of Contents

Project Overview	2
The Need: New Technologies for Information Extraction.....	2
The Research Program.....	2
Mathematical Analysis and Nonlinear Optimization	3
Network Architectures and Design Algorithms.....	3
Application of Research Results.....	3
The Research Team.....	3
Research Results Summary.....	5
Mathematical Analysis and Nonlinear Optimization	5
Summary of Previous Work on Neural Network Theory	5
Introduction to Error Surface Characterization	7
Summary of Results.....	8
Conclusions.....	10
Implications for future work.....	10
Gamma-Nets.....	10
Introduction.....	10
B,D,G phoneme classification	11
Adaptation of delay line order.....	11
Gamma-Delay Lines and MS-TDNN Word Spotting.....	11
Neural Network Cell Image Analysis.....	12
Introduction.....	12
Challenges:	14
Results.....	15
Spline Nets.....	19
Introduction.....	19
Algorithm Research.....	20
Applications.....	21

Decision-Based Nets.....	21
Overview.....	21
Decision-Based Neural Networks.....	22
A New Probabilistic DBNN.....	24
Real-World Applications.....	25
Appendix A: Technical Reports, Jan - May '95.....	27
Appendix B: List of Technical Reports, Oct '92 - Dec '94	

Accession For	
WHS GRAB	<input checked="" type="checkbox"/>
DEC 94B	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/Date	
Availability Status	
Date	Avail. on/for
A-1	Sept. 92

PROJECT OVERVIEW

Contents of this report

This is the final report of a 32-month ARPA funded project entitled "Neural Networks for Interactive Image and Signal Exploitation". The period covered is from October 1, 1992 through May 29, 1995. The report contains an overview of the purpose of the project and a summary of the research results produced during the entire project. Copies of technical papers developed since the last reporting period, Oct-Dec'94, are contained in Appendix A. Technical papers included in previous reports are listed in Appendix B.

The Need: New Technologies for Information Extraction

In the complex world of interactive image and signal exploitation, large amounts of new data arrive every day, much of it requiring immediate attention. It is an analyst's job to scan this information and determine its meaning and significance, i.e., to identify and comprehend the relevant information, if any, contained in the data. Computer-based tools such as interactive browsing, data visualization and automatic pattern recognition are currently being developed to assist the analyst in performing this task. Significant improvements are needed, however, in the core image and signal processing technologies underlying these tools if they are to keep up with ever increasing volumes of data.

Neural networks show great potential for advancing the state of the art of many computational technologies. To enable their use within interactive image and signal exploitation systems, a number of fundamental neural network advances are required. Neural network design must be changed from an arcane art, to an automatic procedure which can be carried out by non-experts. The time it takes to train neural networks must be on the order of minutes, not hours or days, as is currently typical, even on the most advanced engineering workstations. Learning must not be sensitive to the order of presentation of the training data. Neural network image processing must be optimized to maximize the analyst's perceptual performance, rather than to minimize simple numerical measures of image distortion. And finally, the neural networks developed must be implementable on hardware that integrates well with the other computer platforms supporting the analyst.

The Research Program

The neural network limitations described above are diverse yet deeply interrelated. To make significant progress requires a multi-faceted, yet integrated approach combining research in mathematical analysis, nonlinear

optimization, network architectures and design algorithms, and application of research results to significant current problems. Our research program addressed all these areas and produced significant new results in each.

Mathematical Analysis and Nonlinear Optimization

Our program in this area combined both classical analytic theory and numerical experimentation. We explored fundamental issues in neural network theory and settled a long-standing, open question regarding the uniqueness of neural network representations. We developed new theoretical understanding of neural network training by performing millions of computational experiments on problems that aren't amenable to analytic analysis.

Network Architectures and Design Algorithms

Here, our research embraced several classes of networks.

- Gamma-nets: recurrent networks especially useful for processing time-varying stationary and nonstationary signals
- Decision-Based-Nets: networks particularly suited to hierarchical, parallel and distributed processing implementations
- Spline-Nets: generalized multi-layer perceptrons (MLPs) with B-spline interconnect functions

Each of the three research teams described below concentrated its effort in one of these three areas. All three teams focused their research on developing faster, more effective training algorithms that determine the network architecture as part of the training process.

Application of Research Results

Algorithm research comes to fruition when used successfully to solve significant problems. Application of research results was a key element of our program.

Each team refined and validated their algorithm research by working with real applications including: phoneme classification, word spotting, breast cancer tissue classification, satellite image texture classification, electrocardiogram signal analysis, optical character recognition, and face recognition. Results of this work are summarized below and reported more completely in recent technical reports included in Appendix A and in the previously issued reports that are listed in Appendix B.

The Research Team

This has been a joint program among three complementary, collaborating institutions, the David Sarnoff Research Center, Robicon

Systems Inc. (now Katrix, Inc.), and Princeton University. Sarnoff, as host of the National Information and Display Laboratory, develops state-of-the-art systems to meet the government's needs in image and signal exploitation. Sarnoff is a pioneer in real-time neural network-based signal and image processing and perceptual modeling. Katrix is a leader in developing novel neural network architectures and training methods. Princeton University provides expertise in mapping neural network algorithms onto parallel computing platforms and in conceiving novel neural network architectures and algorithms.

RESEARCH RESULTS SUMMARY

Mathematical Analysis and Nonlinear Optimization

Summary of Previous Work on Neural Network Theory

We view neural network training as a map from parameter space to function space. This enables the search for local minima to be recast as a differential geometry problem. Figure 1 illustrates the concept.

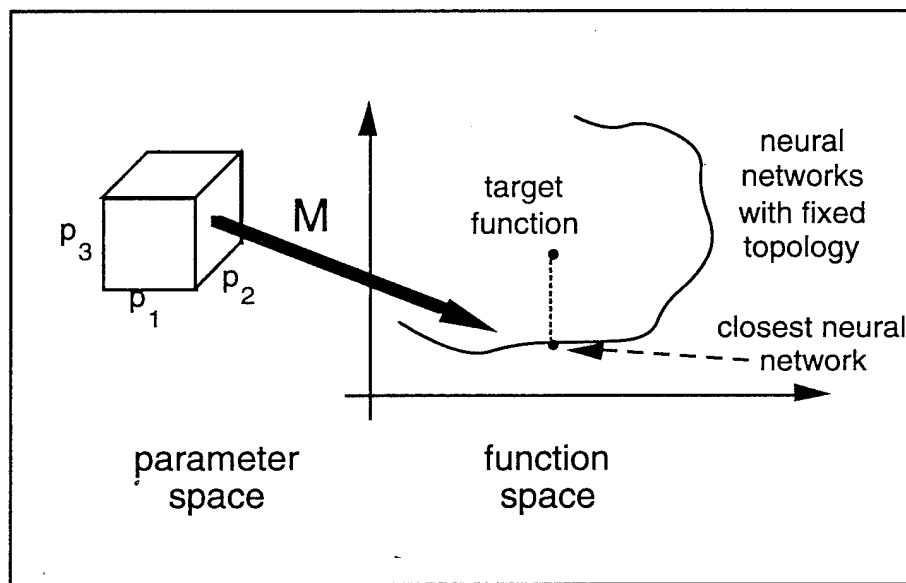


Figure 1. Neural network training viewed as a map M from parameter space to function space.

Figure 2 shows a generic neural network. The map that carries input vectors (x_1, \dots, x_n) to output vectors (y_1, \dots, y_m) is called the output map of the neural network. The key question is: *When do two neural networks have the same output map?*

Charles Fefferman settled this issue by showing that the same map implies the same network [Fefferman, 1994]. The obvious cases of node permutations and sign flips are excluded. This result was obtained using analytic function techniques. Fefferman showed that classical analytic function theory is a valuable tool for understanding the neural network training problem. Specifically, for a given neural network the map ϕ that carries inputs to outputs has an analytic continuation Φ .

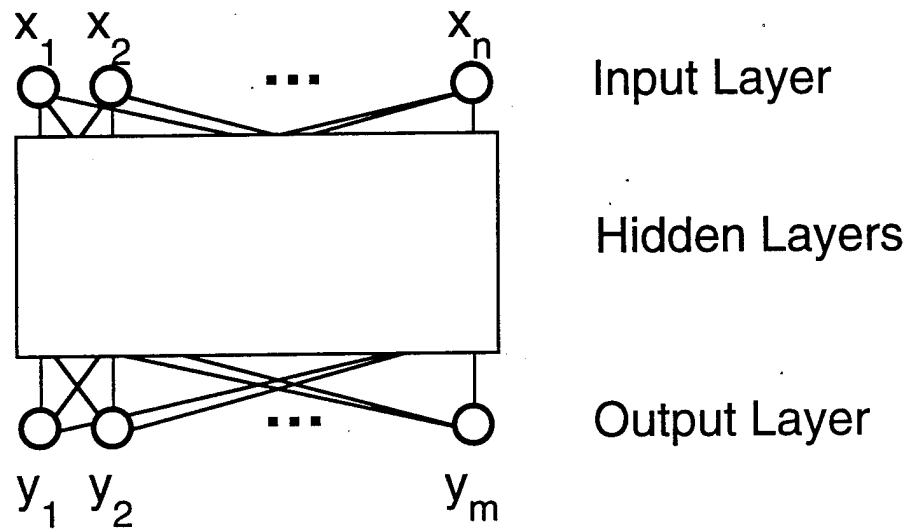


Figure 2. The output map of a neural network.

From the singularities of Φ one can read off essentially complete information about the architecture, weights, and thresholds of the network. Figure 3 shows how this is done for a simple network.

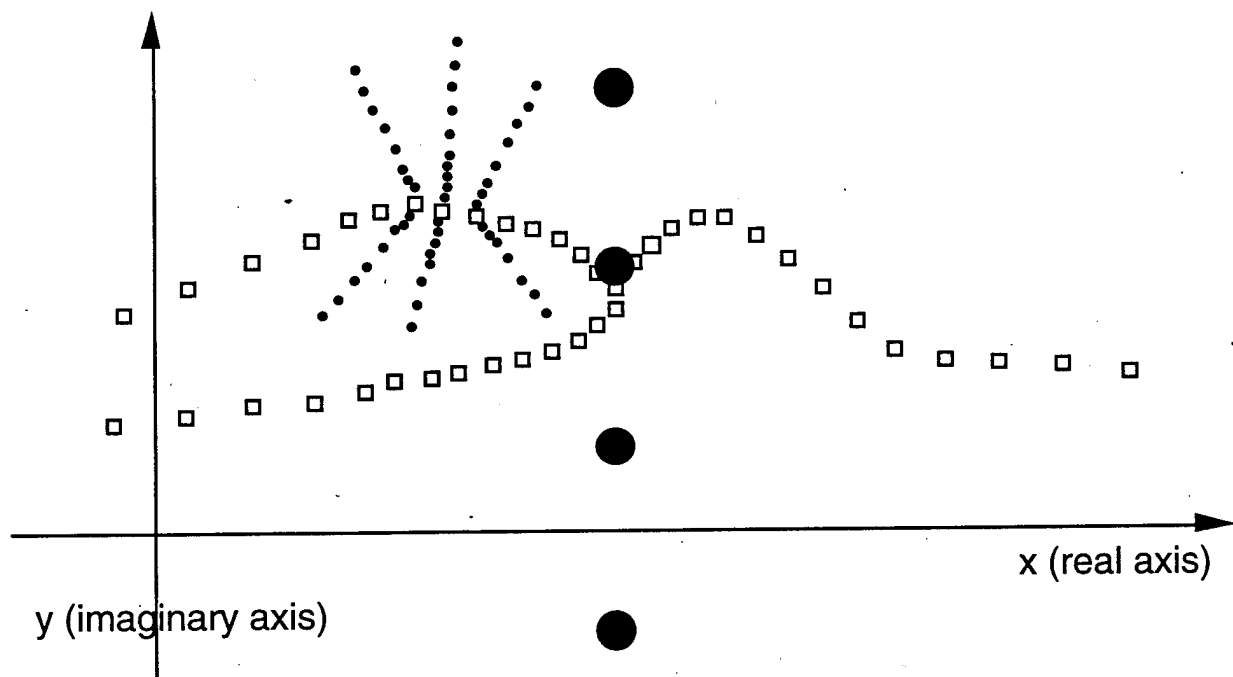


Figure 3. The geometric description of the singularities.

The poles (small dots) accumulate at essential singularities (small squares). Essential singularities (small squares) accumulate at more complicated essential singularities (large dots). The three kinds of singularities (small dots, small squares, large dots) imply three layers of

sigmoids, i.e., two hidden layers and an output layer. The three "spiral arms" of small squares accumulate at each large dot implying three nodes in the second hidden layer. The two "spiral arms" of small dots accumulate at each small square implying two nodes in the first hidden layer. See [Fefferman, 1994] for details.

Another Fefferman result is that the number of critical points (minima, maxima, and saddle points) is finite. See [Fefferman and Markel, 1994].

References

Fefferman, Charles (1994). Reconstructing a neural network from its output. *Revista Mathematica Iberoamericana*. Vol. 10, #3, pp. 507-555.

Fefferman, C. and Markel, S. (1994). Recovering a feed-forward net from its output. *Advanced in Neural Information Processing System 6*, Morgan-Kaufmann, pp. 335-342.

Introduction to Error Surface Characterization

The theme of our experimental work is to develop a rigorous mathematical understanding of neural network objective functions. We describe properties of error surfaces deduced from several million systematic numerical optimization experiments. We ensure that local minima from our experiments are points at which the norm of the gradient is approximately zero, typically less than 10^{-14} , and the Hessian matrix is positive definite. We classify all optimization experiment solutions, not just those that converge to local minima, and compare RMS error distributions for all solutions with those for local minima. A detailed description of our experimental work is included in Appendix A in the paper entitled "Characterizing Neural Network Error Surfaces with a Sequential Quadratic Programming Algorithm".

Although artificial neural networks are being used successfully in many practical applications, their design is still more an art than a science. Statements such as

- Backpropagation works well by avoiding non-optimal minima
- The error surface has many local minima

are common. Little, if any, hard evidence is available to support such claims. We want to replace folklore with facts. In our work, we explore the nature of the error surface, concentrating especially on the question of how many local minima actually exist.

Practical neural networks often have hundreds or thousands of parameters, known as weights. Weights are determined by optimization, called training. While some error surfaces have been characterized for small networks, e.g., the exclusive-or (XOR) network, little can be done analytically to characterize the number of minima for large networks. Thus, we developed the following experimental approach.

- Define a scalable model problem with a known neural network solution with which local minima can be compared
- Perform comprehensive, careful training experiments on modest-sized problems
- Refine training solutions from approximate to highly accurate local minima
- Post-process refinement output and count local minima
- Derive qualitative and quantitative measures describing error surface variation as a function of key network and training parameters

Summary of Results

- Neural network training problems with large amounts of training data (relative to the network size) have few local minima (Figure 4).

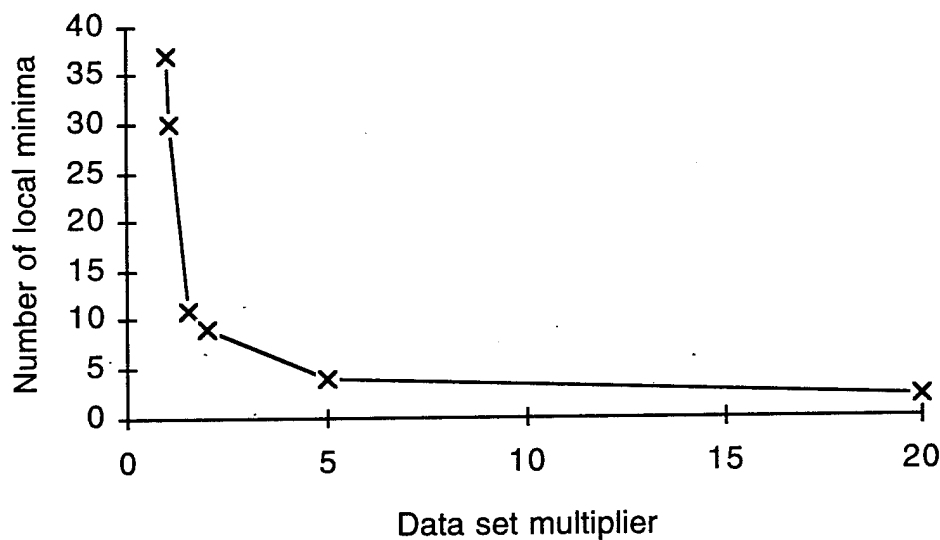


Figure 4: Minima vs. training set size

- Training with small training data sets (relative to the network size) almost never ends by becoming trapped in a true local minimum (Figure 5).

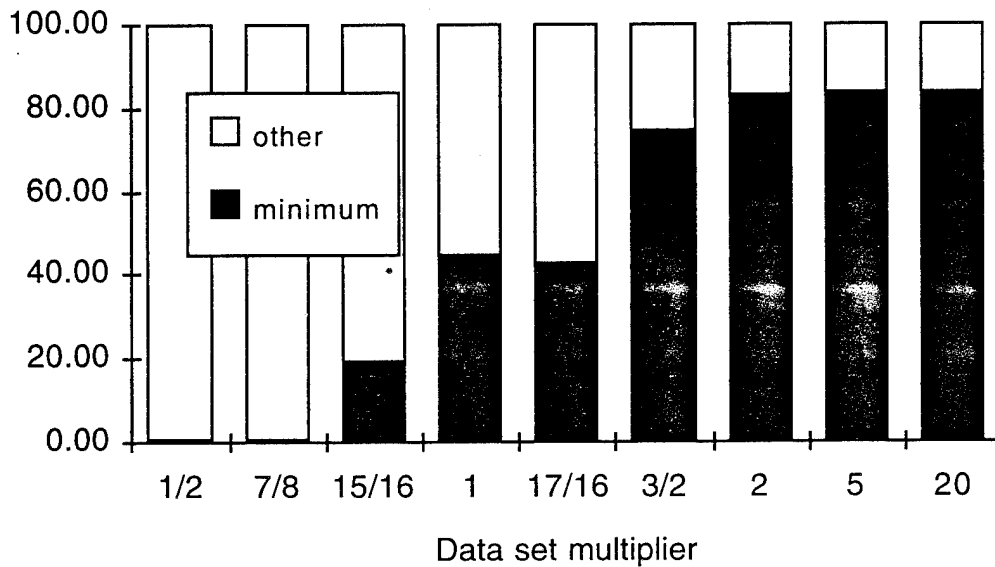


Figure 5: Probability that a random start refines to a minimum

- The number of true minima in a neural network training problem increases as network complexity increases (Figure 6).

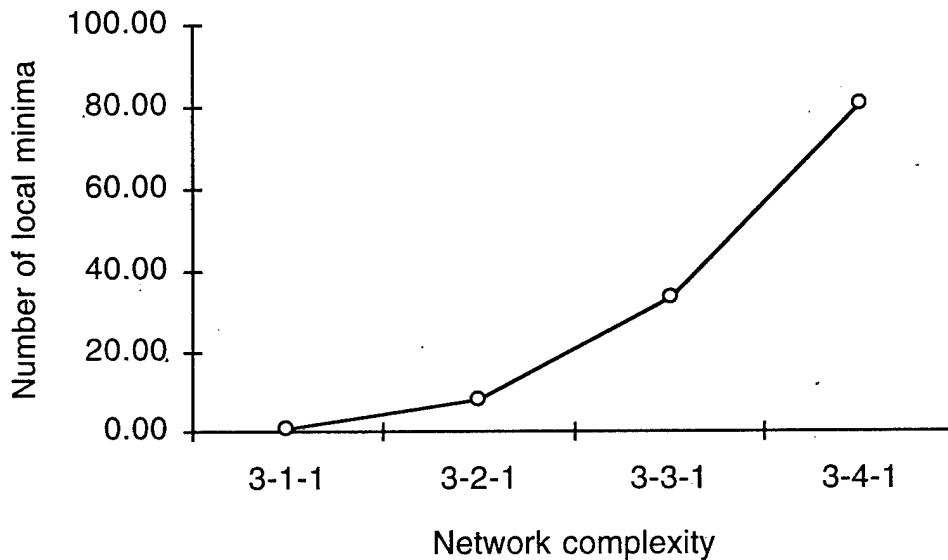


Figure 6: Minima vs. network complexity

Conclusions

To make accurate, reliable statements about the nature of the error surface and the number of local minima requires:

- careful experimental procedures and attention to detail
- high-quality numerical optimization software
- many training runs from randomly generated starts
- refinement and verification of solutions after optimization

When interpreting the results of training experiments, a designer of neural networks for practical applications should take into account whether the problem is data-rich or data-poor. Several, perhaps many, training runs are required to have reasonable confidence that a near-optimal solution has been found.

Implications for future work

Contrary to popular belief, the existence of local minima in the training problem is not a major factor in determining how well a training algorithm performs. This fact could have a significant impact on research and development of faster, more robust training algorithms.

Gamma-Nets

Introduction

The gamma neural net is a model for processing time varying patterns. In the gamma model, the past of a signal is stored in a parameterized tapped delay line of leaky integrators. By adaptation of the parameters of the gamma delay line, an optimal temporal representation can be achieved.

In the framework of this ARPA contract, we have worked on three fronts:

- Application of the gamma neural network to B,D,G phoneme classification.
- Extension of the gamma model to include adaptation of the order of the delay line.
- Application of the gamma network to word spotting.

Here we briefly review the main results of these endeavors.

B,D,G phoneme classification

In 1992, we applied the gamma neural net to the classification of a BDG phoneme set. We used the same data set as was used by Dr. Alex Waibel in his experiments at CMU. Our results indicate that it is possible to save approximately 50% of network size when the gamma delay line is used instead of the tapped delay line. These results were presented at the 1993 GOMAC meeting (de Vries B., Dias L., and Pearson J., An application of Gamma delay lines to "BDG" phoneme classification, Government Microcircuit Applications Conference proceedings, New Orleans, LA, November 1993).

Adaptation of delay line order

A central objective in this research program was to develop adaptive structural algorithms. During most of 1993 we worked on structural adaptation of gamma neural networks. It is becoming very clear that proper network size is fundamental for good generalization performance. For moderate to large problems it is impossible to estimate a priori the appropriate network dimensions, and consequently, automatic network construction is becoming an important research area. Most proposed constructive and pruning algorithms, such as optimal brain surgeon and meiosis nets, make use of a heuristic criterion when deciding if a node or weight should be added or deleted. The method that we developed extends gradient-based learning to the network structure itself. No heuristic decision rule is needed and structure updating takes place alongside weight value updating. A paper on self-structuring delay lines was presented in May 1994 at the International Conference on Artificial Neural Networks in Sorrento (de Vries B., Gradient-based adaptation of network structure, International Conference on Artificial Neural Networks 94, Sorrento, Italy, May 94).

Also, with an eye on speech applications we developed a time-alignment memory filter, which allows for on-line temporal re-alignment so as to compensate for time-warped input signals. These results were presented at the CAIP Neural Network Workshop in New Brunswick and published as a book chapter (Bert de Vries, Short-term memory structures for dynamic neural networks, chapter 5 in Richard Mammone, ed., Artificial Neural Networks for Speech and Vision, Chapman and Hall publ., 1994).

Gamma-Delay Lines and MS-TDNN Word Spotting

During most of 1994, we worked on an NIDL co-funded project whose objective was to improve the performance of the multi-state time-delay neural net (MS-TDNN) word spotting architecture, which had been developed by Carnegie Mellon University (Zeppenfeld, 1992). Allen Reeves (DOD) was the intelligence community sponsor of this project. Our approach was to replace the tapped delay lines in the MS-TDNN with adaptive tapped

delay lines, such as gamma delay lines. In previous works, De Vries et al. (1993) and Renals et al. (1994) showed that the "gamma" enhanced TDNN performs better than a regular TDNN on a phoneme classification task (the /b/, /d/, /g/ set) and a phone classification task on the TIMIT database respectively. In this project we were interested in extending these results to word spotting, and used the Switchboard Speech Corpus, a benchmark speech database for word spotting, which is distributed by the NIST.

The results of the experiments were yet inconclusive; sometimes the system benefited from the added flexibility of adaptive delay lines, but we also reported negative influences from the adaptive delay lines. At this point, we do not see the kind of performance improvements achieved in our previous work with phoneme classification (de Vries et al., 1993).

References

Renals S., Hochberg M., and Robinson T., Learning temporal dependencies in connectionist speech recognition, in Neural Information Processing Systems 6, Cowan, Tesauro and Alspector (eds.), Morgan Kaufmann Publishers, pp. 1051-1058, 1994.

de Vries B., Dias L., and Pearson J., An application of Gamma delay lines to "BDG" phoneme classification, Government Microcircuit Applications Conference proceedings, New Orleans, LA, November 1993.

Zeppenfeld T., Waibel A., A hybrid neural network dynamic programming word spotter, proceedings of ICASSP-92, vol. II, pp. 77-80, San Francisco, CA, 1992.

Neural Network Cell Image Analysis

The following is a summary overview of our complete report, which is included in Appendix A.

Introduction

Cell sample imaging workstations which automate several tedious aspects of individual cell classification are commercially available. An example is one manufactured by Becton-Dickinson (henceforth referred to as the Cell Analysis System, or CAS). This workstation is comprised (essentially) of a microscope, an imaging camera, a computer and related hardware, control software, along with some additional ancillary software programs. It is illustrated, along with the neural network based approach developed in this project, in Figure 1.

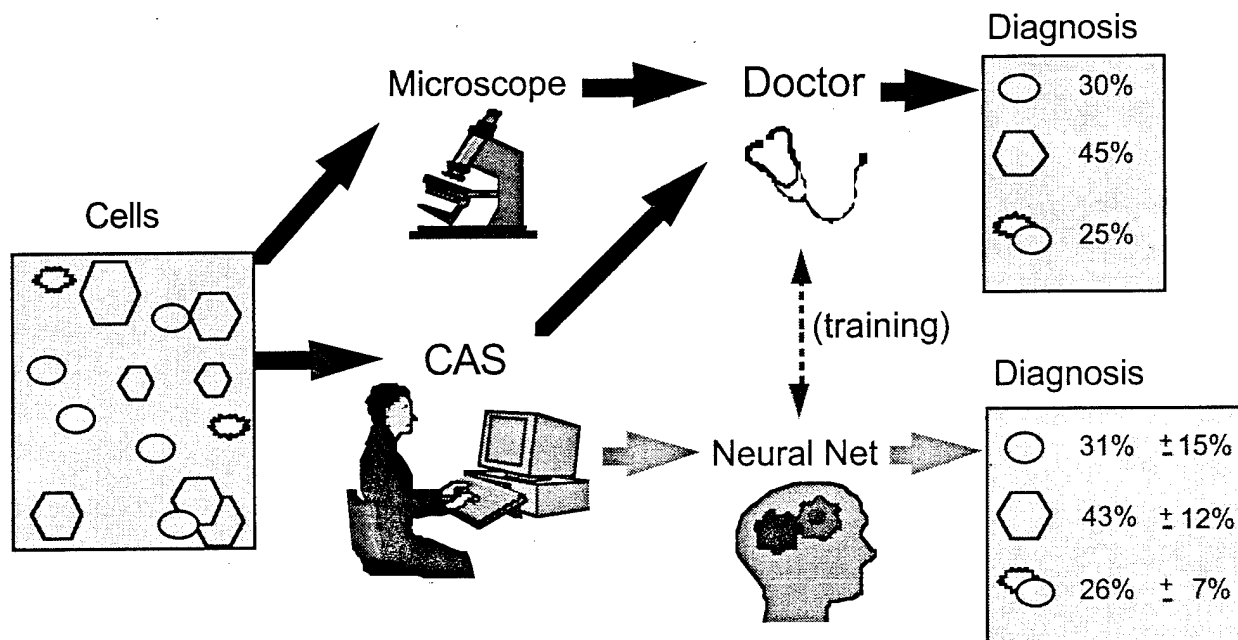


Figure 1. Methodologies for tissue sample analysis

The current state of practice for clinics without a cell imaging workstation is represented by the topmost path: a doctor views a slide preparation with a microscope. The situation where a cell imaging workstation is available is represented by the middle path. A medical technician prepares a tissue sample for analysis (and calibrates the CAS equipment appropriately for the particular sample if necessary). A DNA stain is applied so that only the cell nucleus is visible under the microscope, and so that the pixel intensity is proportional to the DNA density. Under the guidance of a medical doctor, the CAS imaging system analyzes the entire sample. This analysis locates and separates each object in the sample (including each cell nucleus as well as extraneous and irrelevant matter) from the rest of the image, and computes several statistics for each object. The set of objects is passed through a crude filter, classifying each into one of at most 6 classes. The classification is indicated by a visual aid (outlining an image of the cell on a monitor with a certain color). This allows the expert to quickly ascertain the initial classification, as determined by the CAS filters. Then, the medically trained expert reclassifies each object in the sample (by pointing and clicking with a cursor tracking device, i.e., a mouse or trackball). At the same time, the expert deletes any objects that do not correspond to cells with diagnostic potential (e.g., clumped cells, closely touching cells that could not be segmented by the CAS system, broken or damaged cells, extraneous matter, and overlays - i.e., one cell occluding another). If data is at a premium, the expert or the medical technician may further improve on the CAS's image segmentation by manually separating clumped groups of cells into their individual constituents. At present, performing cell classification (with the

maximum amount of automation - i.e., without doing any additional work attempting to improve on the CAS's image segmentation) requires 15 to 30 minutes for slides containing from 100 to 300 cells. There are several aspects of this procedure that can be automated using adaptive computation methods such as neural networks, as is demonstrated by the results of this project.

In conjunction with the Helene Fuld Medical Center in Trenton, New Jersey, we acquired a database of 25,000 cell images. The CAS workstation generated a set of 36 features for each image. This set of features quantify morphometry (size, shape), DNA content (due to the relationship between pixel intensity and DNA density caused by the staining procedure) and a variety of textural measures. We used these features to train neural network models to automate cell image analysis.

The immediate goal of this project was to extend the existing workstation (either with an add-on software program, or by extending the workstation itself) towards achieving two benefits:

1. Automate the detection of "garbage" cells;
2. Improve upon or automate cell classification.

Challenges:

This project posed the following challenges:

1. the need to gather sufficient data to compensate for noise.
2. the need to skew the data gathering process to compensate for the typical class proportions encountered in clinical cases, which are necessarily skewed due to the effect of medical decisions leading up to the determination that a biopsy is warranted.
3. the need to design our methods to be interpretable by the target users, and to be usable in the current clinical working environment.
4. the need to reduce the dimensionality of the feature space for the training examples by judicious selection of especially informative features.
5. the need to train models on a cell-by-cell basis, but to validate the trained models on an entire case (comprised of on the order of 100 to 300 cells).
6. the need to evaluate clinical tissue samples that have much fewer cells than are in tissue samples of other similar approaches (e.g., pap smears have been the target of similar work; a typical pap smear can have from 50,000 to 500,000 cells).
7. the need to provide an estimate of the model confidence.

The latter challenge is particularly important to being able to incorporate these results into an environment where it is necessary to have a measure of certainty associated with the model predictions.

Results

The results are very encouraging. We improved the efficiency of two major aspects of the breast cancer diagnostics process:

1. identifying nondiagnostic objects
2. discriminating between cancer and normal cells.

Here are results of training a 4-output network to discriminate among Normals, Benigns, Cancer, and Nondiagnosics, respectively. The relative frequencies of the classes in the training set are

- 28.6% Normals,
- 24.7% Benigns,
- 25.3% Cancer,
- 21.3% Nondiagnostic.

This network estimated the class frequencies to be

- 28.9% Normals,
- 26.3% Benigns,
- 25.0% Cancer,
- 22.6% Nondiagnostic.

These averages sum to 1.03, and individually are quite close to the actual relative frequencies. Figures 2 and 3 give a more intuitive graphical depiction of the training results, for a 5-output network trained to detect Normal Atypicals, Normal Typical, Benigns, Cancer, and Nondiagnosics.

The following figure shows the distribution of the network outputs on Nondiagnostic in-class examples.

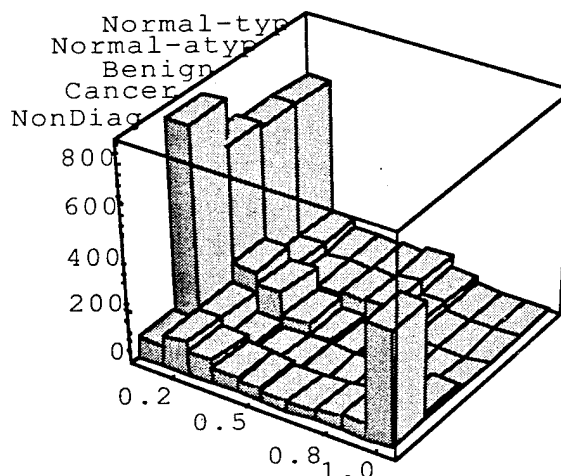


Figure 2. Network distribution when given only Nondiagnostic cells.

It is apparent that this model is good at separating the Nondiagnostic objects. Figure 3 shows the same thing for the Cancer expert.

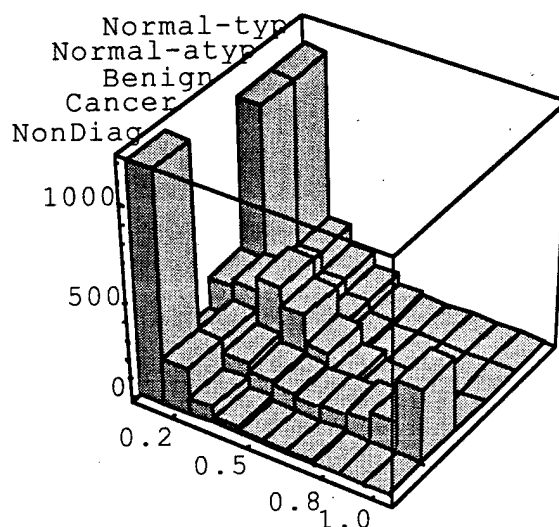


Figure 3. Network distribution when given only Cancer cells.

Examination of the remaining results in the report show that while the Cancer and Nondiagnostic experts did well enough, the Normal and Benign experts did less well. This situation was improved dramatically by using a two-stage approach to first filter the nondiagnostics, and then discriminate between Normals and Cancer. Table 1 presents results for a neural network trained to detect nondiagnostics.

<u>CASES:</u>	<u>NORMAL</u>	<u>BENIGN</u>	<u>CANCER</u>
Actual %:	22	35	32
ECP estimate:	24	29	30
WTA estimate:	14	19	19
%TP:	88	92	69

Table 1. Results for a neural network "expert" trained to detect nondiagnostics. The task is to filter the nondiagnostics from 3 test cases. All values are percentages.

- The "Actual" row gives the actual percentage of each case which is comprised of Nondiagnostics. The next row gives the ECP estimate (estimated conditional probability - the estimated probability of a cell in the case being in the expert's class) of this percentage. The following row gives the WTA estimate (winner-takes-all; for each cell, one expert wins and the others lose, as contrasted with ECP where each expert provides a "soft"

classification for each cell). The last row gives the percentage of examples classified as Nondiagnostic by WTA that actually were Nondiagnostic (i.e., the "True Positives").

Note that some nondiagnostics are missed, and that some cells are incorrectly classified as nondiagnostics. We expect that the situation can be improved by increasing the sensitivity of the Nondiagnostic Expert, thereby resulting in more false positives, and increasing the cell sample size in order to have a sufficient number of cells for stage two (i.e., the Normal vs. Cancer stage).

Table 2 gives the overall error results for the second stage: where two trained "experts" (one for normal, one for cancer) trained to detect normal cells and cancer cells respectively, are applied to each of the 3 test cases after having the nondiagnostics removed.

	Normal case	Benign case	Cancer case
NORMAL "EXPERT":			
1.Actual % :	100	0	4
2.ECP:	88	82	19
3.WTA average:	97 (100)	97	20 (22)
CANCER "EXPERT":			
4.Actual % :	0	0	96
5.ECP:	12	18	81
6.WTA average:	3	3	80 (100)

Table 2: Results of 2 trained "experts" on 3 test cases, with breakdown by expert. The first expert is trained to detect normal cells, the second is trained to detect cancer. Columns give results for each of 3 test cases.

Row titled "Actual %" gives the actual proportion of the cells in the case that fall in the same class for which the "expert" is responsible. Columns in rows 1 and 4 add to 100, except for the benign case, where the cells are neither normal nor cancer.

Each expert gives two types of classification performance: ECP (estimated conditional probability) and WTA (winner-take-all). Columns in rows 2 and 5 add to 100. Columns in rows 3 and 6 add to 100. The numbers in parentheses for the WTA results give the percentage of those that are classified as being within the expert's class that are correctly classified.

Note that the WTA classification was correct 100% of the time for the Normal and Cancer experts given a case dominated by the corresponding class.

Note also that the benign case is diagnosed as "normal" rather than "cancer." This is desirable and intended behavior and will be exploited when developing the third stage in future results, which will discriminate between normal and benign cases.

This cell image dataset is particularly challenging. We found that conventional approaches to separating the data into the labeled class memberships met with varied success (especially when validated in a manner simulating clinical application on 3 test cases). We surmise that there is significant "noise" in the data, because the labeling procedure culminates in data that may have clusters of examples which are nearby in input space while having different class labeling. Compelling evidence for this hypothesis is provided in [Lane & Jim, 1995, Appendix A]. Although not included in our report in Appendix A, we have done additional work since then, applying a nearest neighbor clustering method to the available data, the result of which explicitly demonstrates this effect.

The main result of our research is a 2-stage procedure for automating the discrimination between cancer and normal cells. The first stage of this procedure helps cull out the nondiagnostic objects. The second stage then takes this filtered set and classifies the remaining cells as either noncancer or cancer. The two stage approach is fundamental to the success of the procedure, because different features are useful for detecting nondiagnostic objects than are relevant to discriminating between cancer and noncancerous cells. This is supported by results demonstrated in our complete report (see Appendix A) where a single-stage (i.e., "all-at-once") approach was tried, with limited success. Also see [Lane & Jim, 1995, Appendix A] for another example where the single-stage approach leads to poor generalization results. The two-stage approach resulted in the 3 clinical validation cases being correctly diagnosed.

The results of this project are applicable within the current clinical environment. This is important, because impressive generalization results are largely meaningless to clinical practitioners if they do not simulate a method that can be applied in actual clinical use. Our research demonstrates that these improvements can be implemented without need for extensive retooling of the present working environment. These improvements can be implemented as extensions of a currently available software tool without need for additional hardware. Furthermore, either of these two results can be implemented by retaining the human operator in the loop, or, as a completely automated system. Finally, if implemented as a completely automated system, it is possible for the system to alert the user when appropriate based upon an estimate of model confidence. Note that this level of automation is not possible without an estimate of model confidence. That these encouraging technical results are practically applicable is as important, if

not more so, than the technical results themselves, as this improves the likelihood of these beneficial techniques being incorporated into actual clinical use.

We conclude that this procedure is very promising and warrants further validation. We also expect to find similar merit in an extension of this procedure to multi-stage versions which may result in a procedure for automating not only cancer cell diagnostics, but cancer prognostics, by discriminating between classes of cancer cells grouped according to expected growth patterns. Such an achievement would be instrumental for the evaluation of a particular case, beyond improving early detection (the chances of early detection are improved more by other higher-level methods, such as mammograms), and, given that cancer has been detected, may be useful in determining the most effective means of treatment.

Spline Nets

Introduction

Spline-Nets are part of a larger effort to develop neural network architectures and training methods tailored specifically for image processing and signal exploitation systems. Spline-Nets are generalized versions of Multi-Layer Perceptrons (MLPs) that incorporate B-Spline connection functions into the network node computations in an attempt to trade-off the need for additional hidden layers with increased node complexity. The overall effect is to combine the fast learning and computational efficiency of strictly local network architectures with the scaling and generalization properties of standard MLPs. Research was conducted to extend the Spline-Net architecture and training methods to further enhance their performance in interactive image and signal exploitation problems. Simulation results indicate that the enhancements investigated can reduce training time and mean-square error by a factor of two over previous methods on real-world applications such as Satellite Image Texture Classification and Pathological Cell Classification.

In a standard MLP, the connection functions that relate the output or activation of one node to the input of another is simply a linear function with the slope equal to the value of the weight. Spline-Nets generalize this concept of connection functions by representing them using splines (piecewise polynomials). This enables more powerful connection functions (e.g. piecewise linear, piecewise quadratic, piecewise cubic, etc.) to be formulated, eliminating the need for network architectures with more than one hidden layer. Spline-Nets are initialized with linear connection functions having two knots, one located at 0 and the other located at 1. This makes them equivalent to a standard single-hidden-layer MLP at the beginning of training. During the course of training, additional knots (with corresponding weights) are incrementally introduced into the Spline-Net's

connection functions. This is done using a Bifurcation Schedule which splits in half the intervals between the knots in each connection function, then reinitializes the network weights such that the shape of the resulting connection functions, before and after the split, look the same. The overall result of this type of training is to carve-out coarse global features first, then capture finer and finer localized details later, thereby automatically matching the complexity of the network design to the complexity of the problem. This not only improves the network's generalization capabilities, but reduces the amount of training required to obtain a given level of performance. Use of an appropriate Bifurcation Schedule also simplifies the user interface by making the number of nodes in the single hidden layer the primary design parameter that needs to be specified to obtain optimal network performance.

Algorithm Research

The main goals of this research were to extend the Spline-Net architecture and training methods to further enhance their performance in interactive image and signal exploitation problems. A number of proposed enhancements were investigated to improve Spline-Net learning and generalization while reducing the number of user-specified training parameters. The enhancements investigated included:

- *More powerful learning/training algorithms.* A scheme was developed to adapt Spline-Net connection function learning rates based on current and past local gradient information in an attempt to find the optimal learning rate. The only adjustable parameters in this approach are the minimum and maximum learning rates and the initial learning-rate step size. It has been found that Spline-Net sensitivity to the adaptive learning rate parameters is low, while sensitivity when using fixed-learning-rate "vanilla" backprop is usually quite high.
- *Localized and adaptive Bifurcation Schedules.* The intent of this research was to construct Bifurcation Schedules that perform "optimal" coarse to fine searches in Spline-Net weight space. Bifurcating too late in the training process wastes training iterations as weights oscillate around coarse-scale local minimum. Bifurcating too soon in the training process can lock weights into regions that do not even correspond to coarse-scale local minimum. It was found that Spline-Net Mean-Square-Error (MSE) learning curves could be approximated using a first-order Exponential Decay Model (EDM). This allowed Spline-Net bifurcations to be scheduled as a function of the identified exponential time constant of the model. Simulation results indicate that the EDM approach allowed Adaptive Bifurcation Schedules using output node MSE to reduce training time by a factor of two. Adaptive Bifurcation Schedules based on local measures of MSE at each hidden-layer node were also

investigated. Simulation results indicate that the additional memory and computation required to implement the local Bifurcation Schedules did not produce significant reductions in MSE or training time to warrant their further use.

Applications

Image Texture Classification. (Neural Networks for Classifying Image Textures paper) It was shown that typical textures found in satellite imagery could be correctly classified using a Spline-Net with inputs obtained by preprocessing the data using multi-resolution oriented linear filters to extract relevant high-level features. Only a modest amount of training data, created by hand, was necessary for successful training. The Spline-Net architecture was found to be more effective than a standard MLP of similar topology for this application. The results indicate that many difficult texture classification problems may be solved faster and more efficiently using single hidden-layer Spline Net architectures.

Pathological Cell Classification. Using a Spline-Net with 20 hidden-nodes and an adaptive EDM Bifurcation Schedule, it was found that the CAS database consisting of 25,000 examples of cancerous, benign and normal cells could be classified with an average of 2.0% error in 200 epochs. However, it was found that there is a significant amount of data in and between the training and test sets with high correlated input patterns but conflicting output classifications. As a result, Spline-Net generalization was poor on the test set data, as would be expected. By removing the correlated patterns from the training and test sets, generalization on the test sets is improved. This points to the need to re-examine labelling by the pathologist to check for systematic errors.

Decision-Based Nets

Overview

Neural information processing techniques differ from conventional approaches in their adaptive, nonlinear, and parallel processing characteristics. Inspired by biological vision systems, our research integrates neural network's capability into traditional vision techniques so that systems display the flexibility and reliability closer to what is inherent in biological vision systems. The major advantage hinge upon that they can learn from examples rather than requiring explicit descriptions of concepts they are asked to identify. This in turn offers much greater versatility, reliability, and robustness. Neural networks are amenable to systems with a large number of

processing cells enhanced by hierarchically structured interconnection. Their effective application-specific implementation hinges upon a thorough understanding of hierarchical network structure, training efficiency, real-time retrieving, and parallel processing technology.

In this research, we have successfully developed a "decision-based Neural network" (DBNN), which combined approximation and decision based networks and selected optimal NN structure and basis/discriminant functions. We have also accomplished adaptive integration of hierarchical networks for real-world experimentation on signal/image systems. (We have successfully applied DBNN to a number applications, OCR, texture analysis, ECG, and face recognition.) In terms of recognition accuracy, processing speeds, and parallel processing, the hierarchical DBNN perform far superior to that of the conventional multilayer perceptron (MLP).

As to implementation of neural processing architectures, DBNNs are very amenable to hierarchical, parallel and distributed processing. With respect to implementation of the information architectures, image and vision processors would most likely be implemented as parallel processing architectures. Neural networks are amenable to systems using a large number of processing cells enhanced by extensive interconnectivity. Indeed, the popularity of neural networks owes a lot to the availability of cost-effective massive parallel processing hardware. Fast and parallel digital processors will be of great importance in carrying out research over a longer period with the goal of incorporating powerful hardware employing some appropriate hybrid of digital and analog microelectronics. Mapping algorithms onto parallel neurocomputers is a mature technology (see S.Y. Kung, "Digital Neural Networks", Prentice-Hall, 1993). Groundwork for development and study need to be established on hardware/software codesign, hierarchical processing architecture memory management, neural system synthesis, and hybrid systems combining analog and digital technologies.

Decision-Based Neural Networks

In *hierarchical recognition systems*, the feature representation becomes less explicit and therefore becomes more robust with respect to variations as it ascends in the hierarchy. Our research strives for a closer match between hierarchical image representation/clustering and hierarchical information processing systems. To this end, the DBNN adopts a hierarchical structure with nonlinear basis function. It uses a distributed and localized updating rule based on reinforced and anti-reinforced learning strategy, which allows the border between any two classes to be settled mutually and locally. Such a learning rule, in contrast to that of back-propagation model, incurs minimum side-effects on other borders so as to alleviate the problem of overtraining the networks.

Typically, one output node is designated to represent one class. The All-Class-in-One-Network (ACON) structure is adopted by the conventional MLP, where all the classes are lumped into one super-network. The supernet has the burden of having to simultaneously satisfy all the teachers, so the number of hidden units tends to be large. Empirical results confirm that the convergence rate of ACON degrades drastically with respect to the network size because the training of hidden units is influenced by (potentially conflicting) signals from different teachers. The DBNN adopts a One-Class-in-One-Network (OCON) structure, where one subnet is designated to one class only. Each subnet specializes in distinguishing its own class from the others, so the number of hidden units is usually small. Experimental results based on a broad range of applications (OCR, speech, and face recognition) suggest that 3-5 hidden units per subnet are all it is needed. In these applications, the DBNN's OCON prevails over MLP's ACON in training and recognition speed, especially when the number of classes is very large.

An important aspect of neural net systems concerns the basic functions of neurons. We propose an **elliptic basis function** (EBF) as basis of the discriminant functions. The EBF is similar to RBF except every feature is assigned a different weighting factor which will be part of the learning process. In our extensive simulations, LBF, RBF and EBF networks are compared. EBF has consistently outperformed the other networks in terms of classification rates.

In contrast to traditional MLP (multi-layer perceptron), where the credit-assignments affect all the weight parameters in all the subnets, DBNN adopts a **structurally-selective distributed learning**. There are three main aspects of the distributed DBNN training rule: (1) "When to update": Update weights only when misclassification. (2) "What to update": The learning rule is distributive and localized. It applies reinforced learning to the subnet corresponding to the correct class and antireinforced learning to the (unduly) winning subnet. (3) "How to update": Adjust the boundary by updating the weight vector either in the direction of the gradient of the discriminant function (i.e., reinforced learning) or opposite to that direction (i.e., antireinforced learning). The DBNN adopts a **decision-based learning rule**. When there is misclassification, the correct class which should have won the competition but failed will be applied **reinforced learning** (i.e. its discriminant function gets updated along the greatest gradient ascent direction). At the same time, the class which unduly won the competition will be applied **anti-reinforced learning** (i.e. its discriminant function gets updated along the greatest gradient descent direction).

The training scheme of DBNN is based on the so-called **LUGS** (Locally Unsupervised Globally Supervised) learning. Two levels of the network hierarchy are adopted: In the Global Level: Supervised mutual (decision-

based) learning rule is adopted. In the Local Level: First, the initialization is always by a unsupervised clustering method, e.g. k-mean. If too many clusters are adopted, it often results in overfitting, which in turn will hamper the generalization capability. A proper number of clusters can be determined by unsupervised clustering technique. The first phase is the Locally-Unsupervised (LU) learning phase, during which each subnet is trained individually and no mutual information across the classes is utilized. After the LU phase is completed, the training enters the Globally-Supervised (GS) phase. In GS phase teacher information is used to reinforce or anti-reinforce decision boundaries.

Another approach to adaptively finding optimal hidden units is via a combination of an SVD network and a laterally pruning/growing APEX network. This allows growing or pruning of a model the network whenever an order update is needed in a nonstationary or semi-stationary environment. (see S.Y. Kung, et al. "Adaptive Principal Component EXtraction (APEX) and Applications" Vol. 42, No. 5, May, pp. 1202-1217,, IEEE Transactions on Signal Processing, 1994) This approach offers the advantage of obviating the need of a prior knowledge on the number of hidden units. A new structural approach to node pruning/growing is proposed, by which the appropriate node number in order to classify a given pattern distribution can be estimated. In formulating a structural updating rule, a vigilance test can be applied to the pruning/growing of a laterally connected APEX type network. This helps determine the timing of inducting new neural nodes. The novel structure also facilitates the growing or shrinking of the network whenever an order update is required. The structural similarity between the APEX and the cascade correlation nets offers new insight to a structural optimization analysis. Along a similar line, we have developed a new learning network model as a proper model for transmission through a noisy channel. Based on an intimate relationship between information maximization (in the Gaussian setting) and noisy principal component analysis, we are able to provide an analysis on the sensitivity of noisy PCA solutions and develop new approaches to coping with channel noise.

A New Probabilistic DBNN

The original DBNN's learning is "decision-boundary driven". When pattern classes are clearly separated, such learning usually provides very fast and yet satisfactory learning performance. Application examples including OCR and (finite) face/object recognition. A local winner is the winner among the clusters within the same subnet. The antireinforced learning is applied only to the local winner within the globally winning subnet; and the reinforced learning is applied to the local winner within the correct (and supposedly winning) class. Thus, only the selected clusters are involved in the updating. The boundary can thus be effectively negotiated by two

involving local subclusters. The objective of the training is to find a set of weights which yields a correct classification.

When dealing with overlapping distributions and/or the problem of false acceptance/rejection, it is preferable to adopt the probabilistic DBNN. It is very appealing to applications such as face recognition, false acceptance/rejection, and verification. A new variant of DBNN adopts a probabilistic perspective, where the subnets are designed to model the log-likelihood functions. Reinforced and antireinforced learning is applied to "all" the clusters of the global winner and the supposed (i.e. the correct) winner, with a weighting distribution proportional to the degree of possible involvement (measured by the likelihood) by each cluster.

Recently, we have investigated Expectation-Maximization (EM) for training probabilistic DBNN with applications to multi-channel fusion network. (See S.H. Lin and S.Y. Kung, "Probabilistic DBNN via Expectation-Maximization with Multi-sensor Classification Applications", to appear in Proceedings, Inter. Conf. on Image Processing, Washington, D.C., 1995.) The strategy is to have the probabilistic DBNN approximate the Bayesian posterior probability. EM begins with an optimization of a likelihood function which may be considerably simplified if a set of "hidden" variables are pretended to be known. With generalized EM algorithm, the probabilistic DBNN can potentially offer an improved learning speed and discriminating capability. As supported by experiments, it not only can improve generalization performance, it can also achieve much a lower false acceptance/rejection rate.

Moreover, the multi-sensor classification problems may be handled straightforwardly. The DBNN offers a hierarchy structure which is very amenable to combining various sensor sources. This leads to a hybrid EM and DBNN version of multi-channel fusion network. In a class-dependent channel fusion, we assign a confidence for each channel, which stands for the confidence we have on channel k in class subnet. Note that once the NN is trained, then the fusion weights will remain constant during the retrieving phase. (In the fusion experiment based on intensity and edge features, a class-dependent fusion network achieved 100% recognition of different car models via various viewing angles.) Another innovative hierarchical multi-sensor fusion structure is proposed, where sensors are labeled primary and secondary and they are cascaded in sequential processing stages. In our experiment, the hierarchical multi-sensor fusion has substantially reduced false acceptance/rejection rates.

Real-World Applications

DBNN can be applied to multiple information processing levels, from feature extraction to object recognition and scene analysis. (See S.Y. Kung and J.S. Taur, "Decision-based Hierarchical Neural Networks with Signal/Image

Classification Applications." IEEE Transactions on Neural Networks, Vol. 6, No. 1, pp. 170- 181, January 1995.) To achieve high accuracy, we have observed a critical need for a robust facial feature. To enhance DBNN training, under a class of defined invariant space, we elect to transform the original training set to create additional training exemplars. The newly generated set may be considered as virtual training patterns.

Various real application experiments have been performed, including ECG signal analysis, OCR, and texture classification. Two types of patterns must be differentiated: (a) *static patterns* and (b) *temporal patterns*. Static patterns are not order-sensitive, while *temporal patterns have strong order sensitivity*. The temporal model must be chosen so as to adequately manifest the vital temporal characteristics and remains robust with respect to the unpredictable temporal variations such as shift or warping. By incorporating a temporal discriminant function into the Decision-Based Neural Network, we have developed a prediction-based independent training networks. We are applying them to ECG classification application and investigate the tolerance of temporal misalignment of ECG waveforms. As examples of static pattern classification, the DBNN performs extremely well in texture-classification and OCR applications. For texture classification, the training performance is 100% while the test accuracy 97%. The conventional approximation-based BP method has very slow speed and persistently large mean-squares error. The OCR problem experimented with is to recognize a rectangular pixel array as one of the 26 capital letters in the English alphabet. Holland-style adaptive classifiers obtains only an accuracy of 80%. The training accuracy of DBNN is 99.9% and the generalization accuracy is above 92%. In fact, our approach obviates the need of a prior knowledge on the higher order statistics of the textures, since the appropriate features to classify various textures can be learned through training the network weights. The DBNN has been successfully applied to several face recognition experiments recognizing up to 200 people. As part of feature extraction, warped face images will first be intensity-normalized and resolution-reduced to provide robust features for DBNN. The DBNN has reportedly achieved very high recognition accuracy with fast training and retrieving speeds. Furthermore, the new probabilistic DBNN allows confidence measures to be incorporated into training neural networks and fusion of different sensors. This substantially improves generalization performance, making DBNN more appealing to real application requirements.

APPENDIX A

Technical papers, Jan-May, 1995

1. Crane, R.L., C. Fefferman, S.A. Markel, and J. Pearson, "Characterizing Neural Network Error Surfaces with a Sequential Quadratic Programming Algorithm", pending submission to SIAM Journal on Optimization.
2. Plutowski, M.E., "Automating Breast Cancer Detection by Neural Network Cell Analysis", Parts 1, 2, and 3, May, 1995.
3. Lane, S.E. and K. Jim, "Classification of Cancer Cell Data using Spline-Nets", July, 1995.

Characterizing Neural Network Error Surfaces with a Sequential Quadratic Programming Algorithm

Roger Crane, Charles Fefferman*, Scott Markel, and John Pearson

David Sarnoff Research Center

CN 5300

Princeton, NJ 08543-5300

rcrane@sarnoff.com

cf@math.princeton.edu

smarkel@sarnoff.com

jpearson@sarnoff.com

* Alternate address: Dept. of Mathematics, Princeton University, Princeton, NJ 08544

ABSTRACT

We describe properties of error surfaces deduced from several million systematic numerical optimization experiments. We ensure that local minima from our experiments are points at which the norm of the gradient is approximately zero, typically less than 10^{-14} , and the Hessian matrix is positive definite. We describe our experimental methodology in sufficient detail to demonstrate that our results are reliable.

Our principal results show:

- how the number of minima varies as a function of the ratio of training set size to network size;
- the probability that training from a random start converges to a minimum
- how the number of minima increases with network complexity.

We classify all optimization experiment solutions, not just those that converge to local minima, and compare RMS error distributions for all solutions with those for local minima.

TABLE OF CONTENTS

Introduction.....	1
Folklore.....	1
Goal of our research.....	1
Approach.....	1
Summary of principal results	2
Experimental methodology.....	4
Methodology Overview	4
Defining a model problem.....	4
Solving many training problems.....	4
Post-processing the results from training.....	4
Methodology Details	5
Defining the model problem.....	5
Solving training problems.....	5
Generating targets and starting points.....	5
Training to convergence.....	7
Post-processing the results.....	8
Refining candidate solutions.....	8
Eliminating topologically equivalent solutions.....	9
Counting local minima.....	10
Classifying candidate solutions that fail to refine.....	10
Validity of results.....	12
Error surface Dependence on Training Set Size.....	13
Experiments run for 3-3-1 networks.....	13
Linear theory.....	13
Relation between number of minima and number of random starts (DSM \approx 1)	13
Relation between number of minima and number of random starts (DSM \geq 1)	15
Distribution of RMS errors for refined solutions (DSM = 1).....	15
Distribution of RMS errors for exemplars (DSM = 1).....	16

Classifying all candidate solutions	18
Probability of refinement termination.....	18
Distribution of RMS errors for all solutions.....	20
Distribution of RMS errors for exemplars.....	22
Error surface dependence on network complexity	24
Experiments run for 3-N-1 networks	24
Relation between number of minima and number of random starts (DSM = 1)	24
Increase in minima with network complexity.....	26
Conclusions.....	28
Error surface dependence on the Data Set Multiplier.....	28
Minima and network complexity	28
Experimental Methodology	28
Limitations of our approach.....	29
Implications for solving application problems.....	29
Future work.....	29
References.....	31
Acknowledgments.....	32
Appendix	33
Glossary	33

INTRODUCTION

Note: The Appendix contains a glossary of terms for quick reference. Most terms specific to this report are also defined when they are used first.

Folklore

Although artificial neural networks¹ are being used successfully in many practical applications, their design is still more an art than a science. Statements such as

- Backpropagation works well by avoiding non-optimal minima
- The error surface has many local minima

are common. Little, if any, hard evidence is available to support such claims.

Goal of our research

We want to replace folklore with facts. We then intend to develop efficient, reliable neural network design methods based on sound principles. In this paper, we explore the nature of the error surface, concentrating especially on the question of how many local minima actually exist.

Approach

Practical neural networks often have hundreds or thousands of parameters, known as weights. Weights are determined by optimization, called training. While some error surfaces have been characterized for small networks, e.g., the exclusive-or (XOR) network [Hamey, 1995], little can be done analytically to characterize the number of minima for large networks. Thus, we developed the following experimental approach.

- Define a scalable model problem with a known neural network solution with which local minima can be compared
- Perform comprehensive, careful training experiments on modest-sized problems

¹Introductory material on neural networks can be found in [Hertz, Krogh, and Palmer (1991)]. Chapters 5 and 6 are especially relevant to this report.

- Refine training solutions from approximate to highly accurate local minima
- Post-process refinement output and count local minima
- Derive qualitative and quantitative measures describing error surface variation as a function of key network and training parameters

Summary of principal results

The following results hold for fully-connected networks with one hidden layer:

- For a fixed network topology, the number of local minima is a decreasing function of the ratio of the training set size to the network size (number of weights). (Figure 1)

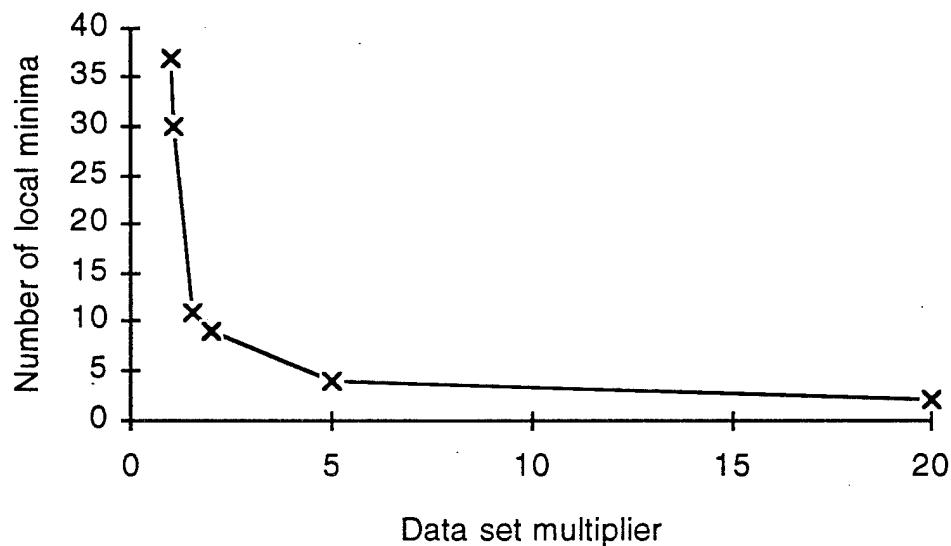


Figure 1: Minima vs. training set size

- When the training set size is less than the network size, it is unlikely (i.e., there is a low probability) that training a random start followed by refinement leads to a local minimum. (Figure 2)

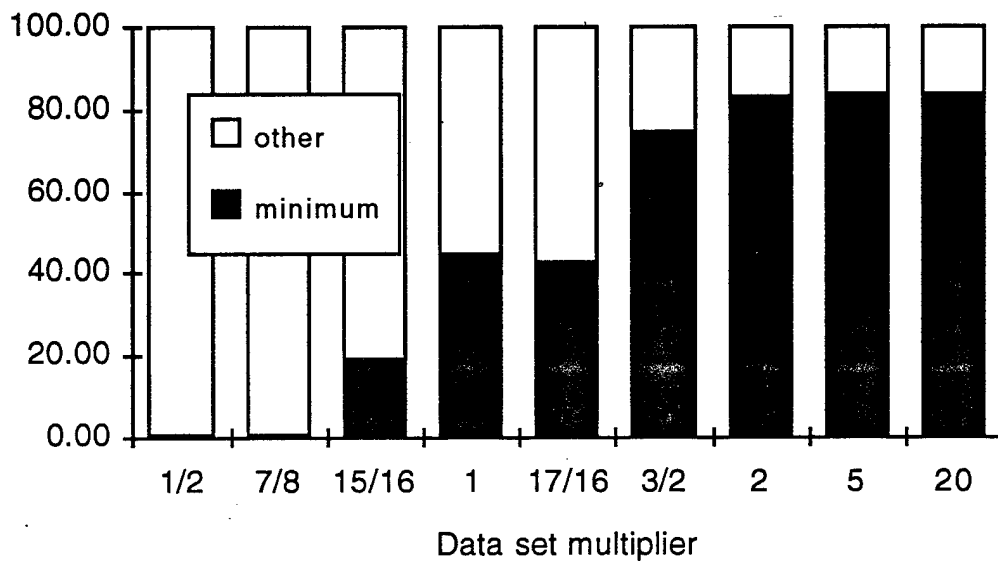


Figure 2: Probability that a random start refines to a minimum

- For a fixed ratio of training set size to network size, the number of local minima increases with the network size. (Figure 3)

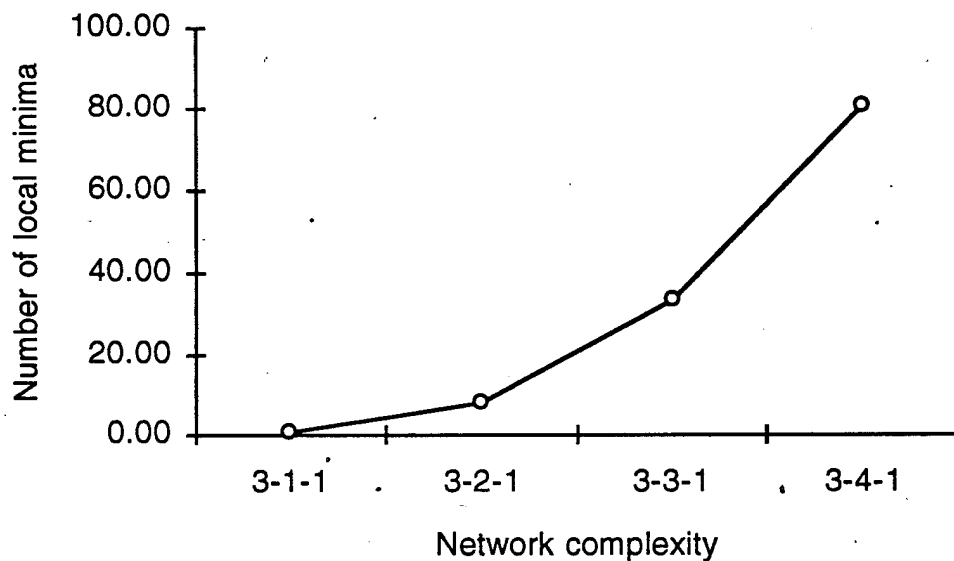


Figure 3: Minima vs. network complexity (DSM = 1)

A related figure (Figure 14) that includes supplementary information about the variance, appears later in this report.

EXPERIMENTAL METHODOLOGY

Accurately identifying local minima in an experimental setting requires both careful attention to detail and high-quality numerical algorithms and software. We next describe our experimental procedures, first with an outline of the major steps, then by elaboration of the steps.

Methodology Overview

Our experiments use model training problems. Our target is a neural network, i.e., the training problem is constructed so that it is solved exactly by a known neural network. By using a model problem, starting training from a point in weight space far from the known solution, and converging to the known solution, we verify that our numerical procedures are robust and accurate. We gather statistics about other minima by using many starting points.

Experiments entail:

Defining a model problem

- Picking a network topology
- Generating random input data and propagating it through the network
- Using the resulting input-output pairs for training data

Solving many training problems

- Generating random starting points
- Finding candidate solutions by optimizing to convergence

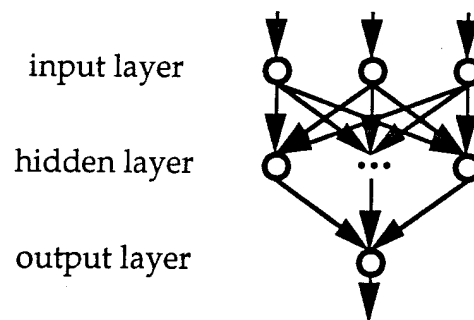
Post-processing the results from training

- Refining candidate solutions to highly accurate local minima
- Eliminating topologically equivalent solutions
- Counting the number of local minima
- Classifying the candidate solutions that fail to refine

Methodology Details

Defining the model problem

Our studies reported here use fully connected 3-N-1 networks, i.e., networks with 3 input nodes, one hidden layer with N nodes, and one output node.



Hidden and output nodes have odd sigmoidal function nonlinearities. Choosing N specifies the topology.

Solving training problems

Generating targets and starting points

We call the network that solves a model training problem exactly a target network. The target corresponds to a global minimum of the training problem, and the minimum error is zero. If there were no other minima, an ideal training algorithm would always converge to the target. A starting point is a network whose parameters (weights) are used to start a training (optimization) run.

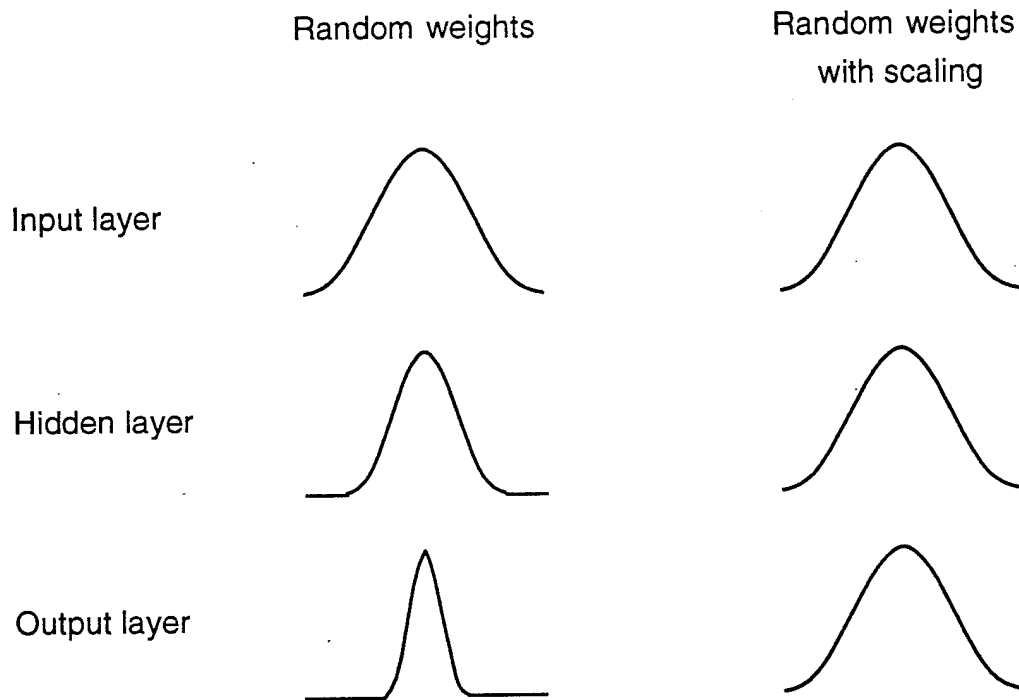
Targets and starting points are both generated with the following 4-step algorithm.

- 1) Assign random network weights
- 2) Generate normally distributed random input
- 3) Propagate the input data through the network layer by layer
- 4) Scale the initial random weights to preserve, approximately, input data distributions at all nodes

Algorithm details follow.

We developed a constrained, random procedure to generate well-conditioned targets and starting points. By approximately preserving the input data distribution at each network node we ensure that no node is

saturated, i.e., that no node output is constant. Saturation occurs when operating at the extremes of the nonlinear sigmoidal function. The following diagram illustrates another motivation for using the constrained procedure. As random data are propagated through multiple network layers, the width of the distribution narrows. Of course, this is not a serious problem for networks with only one hidden layer.



Algorithm for generating constrained random targets and starting points

Let μ' and σ' be the mean and standard deviation of N (typically 1000) normally distributed input vectors. In our experiments, $\mu' = 0$, $\sigma' = 1$. Proceed layer by layer, from the first hidden layer to the output layer, to assign network weights as follows. For each node:

- Choose initial random weights $\{w_j\}_{j=1}^M$, uniformly distributed on $(-1, 1)$
- Choose random values of μ'' and σ'' such that $\mu'' \approx \mu'$ and $\sigma'' \approx \sigma'$

For targets, $\mu'' \in (-0.1, 0.1)$, and $\sigma'' \in (0.8, 1.2)$. For starting points, $\mu'' = 0$, and $\sigma'' = 1$.

- Let $\left\{ \left\{ y_j^d \right\}_{j=1}^M \right\}_{d=1}^N$ be the output sets from the previous layer

$$\mu = \frac{1}{N} \sum_d \left[\sum_j s w_j y_j^d - \theta \right] \quad \sigma^2 = \frac{1}{N-1} \sum_d \left\{ \left[\sum_j s w_j y_j^d - \theta \right] - \mu \right\}^2$$

- Solve for s and θ such that $\mu = \mu''$ and $\sigma = \sigma''$
- Scale the weights by s

Training to convergence

We used a commercially available Sequential Quadratic Programming (SQP) algorithm [The Numerical Algorithms Group Inc., 1991] to solve training problems. This algorithm is efficient in objective function calls and allows use of parameter (weight) bounds, and linear and nonlinear constraints. A brief description of the algorithm follows. More details are available in the reference cited.

Numerical Algorithms Group (NAG) Library Subroutine E04UCF

E04UCF solves the following problem.

$$\underset{\mathbf{x}}{\text{Minimize}} \quad F(\mathbf{x}) \quad \text{subject to} \quad \mathbf{l} \leq \begin{bmatrix} \mathbf{x} \\ \mathbf{A}_L \mathbf{x} \\ \mathbf{c}(\mathbf{x}) \end{bmatrix} \leq \mathbf{u}, \quad \text{where}$$

$F(\mathbf{x})$, the objective function is, in general, nonlinear, \mathbf{x} is an n element parameter (weight) vector, \mathbf{A}_L is an n_L by n constant matrix, $\mathbf{c}(\mathbf{x})$ is an n_N element vector of nonlinear constraint functions, and \mathbf{l} and \mathbf{u} are constant lower and upper bound vectors, respectively.

E04UCF's major iterations generate a sequence of iterates, $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \mathbf{p}_i$, that converge to a local optimum. The scalar, α , is a step-length, and the search direction, \mathbf{p}_i , is determined by solving a quadratic programming subproblem as defined below. Each solution for \mathbf{p}_i is called a minor iteration. Lagrange multiplier estimates are computed and the active set of constraints is predicted in each major iteration.

$$\text{Minor iteration:} \quad \underset{\mathbf{p}}{\text{minimize}} \quad \mathbf{g}^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \mathbf{H} \mathbf{p} \quad \text{subject to} \quad \bar{\mathbf{l}} \leq \begin{bmatrix} \mathbf{p} \\ \mathbf{A}_L \mathbf{p} \\ \mathbf{A}_N \mathbf{p} \end{bmatrix} \leq \bar{\mathbf{u}},$$

where \mathbf{g} is the gradient of F , the matrix \mathbf{H} is a positive definite approximation of the Hessian of the Lagrangian, \mathbf{A}_N is the Jacobian matrix of \mathbf{c} evaluated at \mathbf{x} , and the upper and lower constraint bounds are simple, additive functions of \mathbf{l} , \mathbf{u} , and \mathbf{x}_i .

Convergence criteria

E04UCF terminates successfully if:

- 1) the sequence of iterates converges, and
- 2) the final point is a Kuhn-Tucker point, i.e., it satisfies the following first-order necessary conditions
 - All constraints are satisfied
 - The gradient can be expressed as a linear function of the Lagrange multipliers
 - Lagrange multipliers corresponding to inequality constraints satisfied at equality have the correct sign

No sufficiency conditions are checked in E04UCF, or any other readily available optimization algorithm known to the authors.

E04UCF options used

Our initial experiments used linear constraints to eliminate topologically equivalent solutions. Using constraints for this purpose didn't work well since, in most cases, the algorithm converged to solutions on constraint boundaries, not to interior local minima. All results reported here were run without constraints except that weights were bounded in $[-8, 8]$.

E04UCF has many control parameters whose default settings are chosen, in general, to achieve maximum numerical accuracy. Default values were used for all, except for the major iteration limit which was set to 150 times the number of parameters (weights) being optimized. With this choice, E04UCF converged for over 99.99% of our training experiments.

Post-processing the results

Refining candidate solutions

The goal of refinement is to pass from an approximate local minimum generated by SQP to an "exact" local minimum, computed to machine precision. For refinement, we used Newton's method and a local quadratic approximation of the objective function.

As mentioned above, SQP solutions satisfy necessary, not sufficient, conditions for a minimum. To identify the true local minima, we attempted to refine all SQP solutions. When Newton's method converged within 50 iterations, the converged (refined) solution was accepted as a local minimum.

Solutions that did not converge were classified and analyzed further, but were not counted as local minima.

We developed an efficient algorithm for computing the exact Hessian, i.e., we did not use a finite difference approximation. Other algorithms are available [Bishop, 1992; Buntine and Weigend, 1993]. Our method requires four propagation passes, two forward and two backward. The number of propagation passes is independent of the network architecture.

Let E , x_i^l , y_i^l , and w_{ij}^l denote the mean squared error, node inputs, node outputs, and weights, where l is a layer number, i is a node number in layer l , and j is a node number in layer $l-1$. The four propagation passes calculate

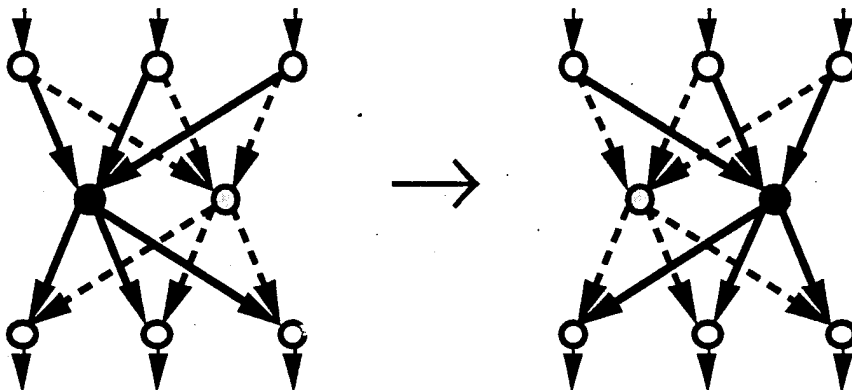
1. node outputs, y_i^l , - forward propagation
2. node deltas, $\frac{\partial E}{\partial y_i^l}$, - back propagation
3. node input mixed partials, $\frac{\partial x_k^n}{\partial x_i^l}$, forward propagation
4. second derivatives, $\frac{\partial^2 E}{\partial w_{ij}^l \partial w_{km}^n}$, back propagation

Refined solutions are guaranteed to be true minima. The iterative refinement process converges if and only if successive iterates differ by less than $\epsilon = 10^{-10}$ (l^2 norm) and the Hessian is positive definite.

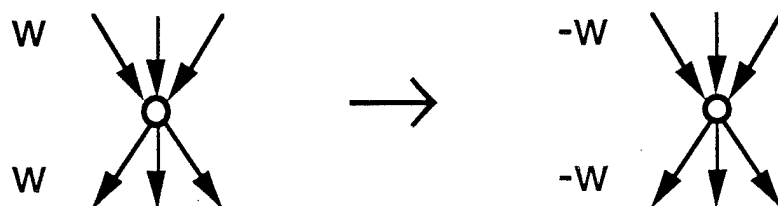
Eliminating topologically equivalent solutions

Two obvious examples of topologically equivalent solutions are shown below.

- Permute the nodes in a hidden layer



- For a hidden node with an odd transfer function, change the sign of each weight (including the bias weight) associated with the node



Topologically equivalent networks can be avoided by reducing networks to canonical form. We used the form referred to in [Chen, Lu, and Hecht-Nielsen, 1993] as a wedge and defined there by a simple set of linear inequalities in the weights. For $M-N-1$ networks, there are $(N!)(2^N)$ possibilities. Factors of the same form would result for each additional hidden layer. With multiple hidden layers, the product of all such terms gives the total number of topologically equivalent networks.

Counting local minima

Local minima are determined by further processing the set of solutions that refine. After transforming refined solutions to canonical form, clustering is used to aggregate equivalent solutions. Clustering refined solutions is simple and accurate since equivalent solutions differ by less than 10^{-10} , the refinement error tolerance. Networks are considered to lie in different clusters if the distance between them is at least 10 times the error tolerance. An exemplar is chosen arbitrarily from the solutions in each cluster to represent the local minimum corresponding to the cluster.

A check is made to verify that the distance between exemplars is at least 5×10^{-9} . Our experiments show that clusters are well-separated. Typically, the distance between exemplars is between 0.1 and 10, approximately 10 orders of magnitude greater than the upper bound for the distance between solutions in the same cluster.

Classifying candidate solutions that fail to refine

Refinement fails if, during the process of iterating in Newton's method, any one of the following occurs:

- 1) a non-positive definite Hessian matrix results
- 2) convergence is not achieved in 50 iterations
- 3) Hessian eigenvalues can't be computed accurately

Root mean square (RMS) error distributions for solutions in each of these three categories are compared below with error distributions for solutions that refined successfully.

VALIDITY OF RESULTS

For the model problem we studied, our results are reliable for the following reasons.

- Computations were made in double precision and thoroughly tested
 - We verified consistency among our objective function, gradient, and Hessian calculations by using the derivative checker provided in the NAG library.
- Training with SQP converged for over 99.99% of the experiments run
 - We used the NAG E04UCF algorithm default accuracy parameters, which provide, in general, the maximum numerical accuracy possible. Three separate tests for convergence are made. With these stringent conditions, a convergence rate of 99.99% is convincing.
- We ran sufficiently large experiments
 - Several graphs showing the dependence of minima counts on experiment size (number of random starts used) are presented in the results below to support this claim.
- Refinement permits unambiguous clustering
 - Clustering is simple and reliable since refinement resolves minima (in weight space) to less than 10^{-10}
 - An exemplar can be chosen arbitrarily from each cluster since solutions in the same cluster are essentially identical
 - Distance between two exemplars (in weight space) is typically between 0.1 and 10, i.e., clusters are well-separated
 - We include in the next section a histogram showing a marked difference in the distribution of RMS errors before and after refinement. While the performance of SQP is excellent, refinement is necessary to accurately count local minima and analyze error distributions.
- Our experimental results agree with a theoretical prediction
 - For linear networks, elementary linear algebra shows that the character of the solution changes dramatically as a function of training set size. We present results below that exhibit an analogous effect for nonlinear neural networks.

ERROR SURFACE DEPENDENCE ON TRAINING SET SIZE

Experiments run for 3-3-1 networks

To interpret the following results requires understanding what is meant by a Data Set Multiplier (DSM). The DSM is a ratio; the number of input-output pairs in the training set / number of network weights. When the $DSM = 1$, for example, a training problem has the same number of data points (input-output pairs) to be fit as unknown weights to be determined by optimization and refinement.

A 3-3-1 network has 16 weights, 4 of which are called bias weights or thresholds. The table below shows the multipliers for which experiments were run and, for each, the corresponding number of data points. 20K random starts and 1 target were used for each DSM, a total of 180K training runs.

DSM	1/2	7/8	15/16	1	17/16	3/2	2	5	20
# data points	8	14	15	16	17	24	48	80	320

Linear theory

A linear neural network can be created by replacing the nonlinear sigmoidal functions in each hidden and output layer node by linear functions. Solutions for such a system have the properties listed below.

- A linear, non-degenerate system with $DSM \geq 1$ has a unique least squares solution
- A linear system with $DSM < 1$ has an infinite number of solutions

Relation between number of minima and number of random starts ($DSM \approx 1$)

Figure 4 shows how the number of minima found in our experiments increases as the number of random starts increases.

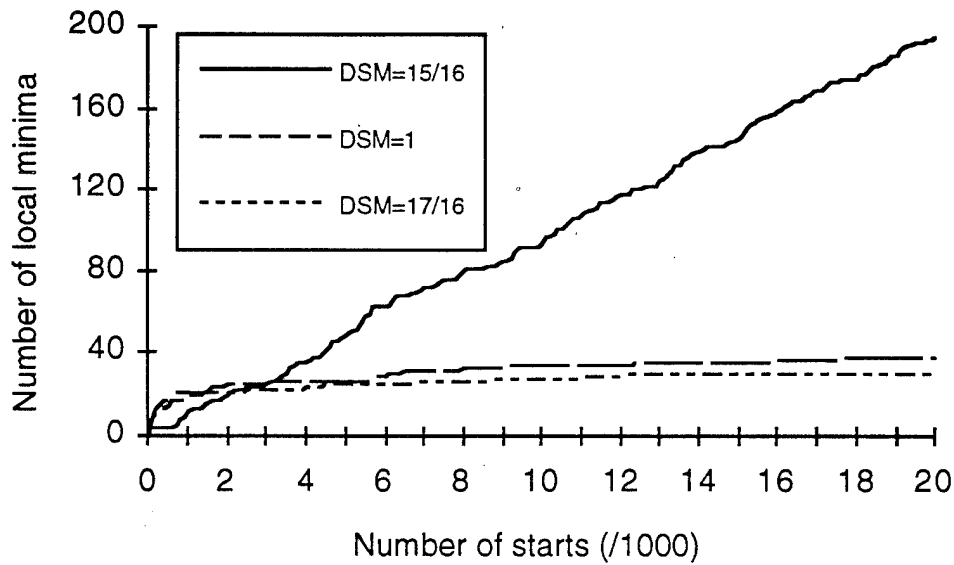


Figure 4: Minima vs. number of starts

- The qualitative change in behavior is obvious for $DSM < 1$ vs. $DSM \geq 1$
- The number of minima increases linearly for $DSM = 15/16$

In addition to these observations, the numbers of minima for $DSM = 1$ and $DSM = 17/16$ exhibit asymptotic behavior as the number of starts approaches 20,000. Thus there is a low probability of finding significantly more minima by increasing the experiment size (number of starts). In contrast, the number of minima increases without bound for $DSM = 15/16$. This is not surprising, since the corresponding linear problem has an infinite number of solutions.

Relation between number of minima and number of random starts (DSM ≥ 1)

Figure 5 shows results for all DSM ≥ 1 cases run.

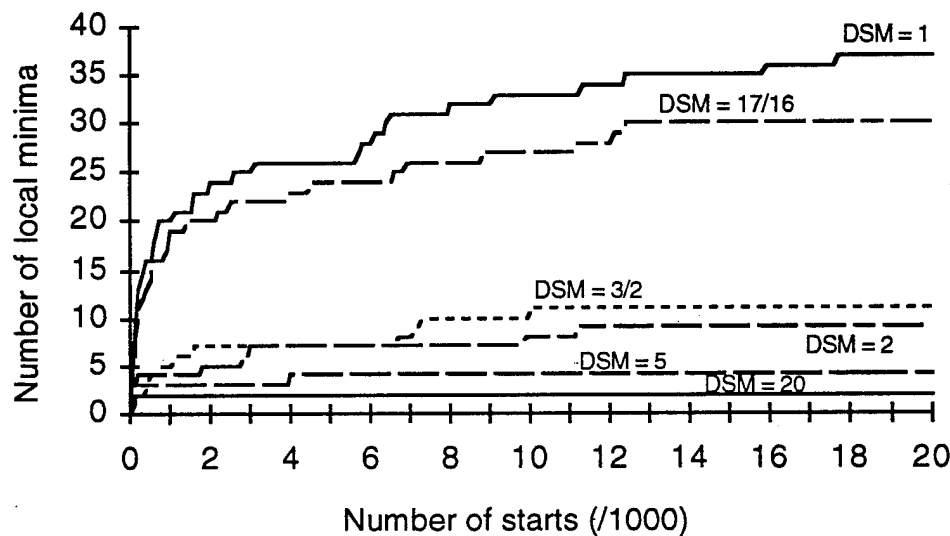


Figure 5: Minima vs. number of starts

In each case, the number of minima approaches an asymptote. 20K starts are sufficient since few, if any, additional minima would be found by using more starting points.

- Number of minima decreases with increasing DSM
- Very few minima for large DSMs

These facts can be deduced from Figure 5, but are more clearly evident in Figure 1 (see Introduction).

Distribution of RMS errors for refined solutions (DSM = 1)

Recall that 20K starts were used in these experiments. For the DSM = 1 experiment shown in Figure 4, 44% (8802) of the random starts refined to a local minimum (See Figure 2). Figure 6 compares results before and after refinement for all solutions that refined successfully.

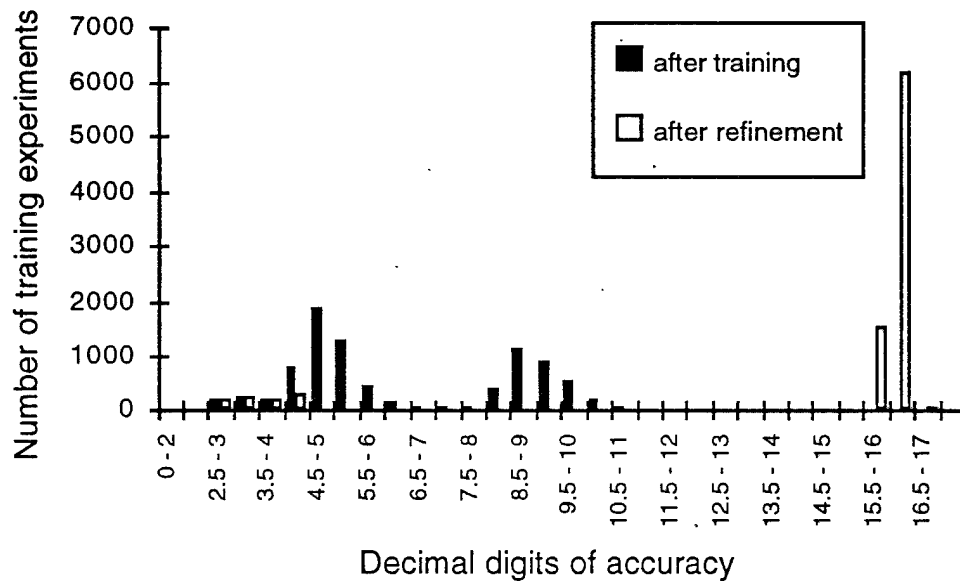


Figure 6: Distribution of RMS errors for all refined solutions (DSM = 1)

- Refinement has a striking effect on the distribution of RMS errors
 - No SQP solution (before refinement) has an RMS error near 10^{-16}
 - Most (7775/8802) refined solutions have RMS errors near machine precision (approximately 10^{-16})
- Errors of 10^{-16} closely approximate the known global minimum.
- A significant number (1027/8802) of refined solutions have errors between 10^{-2} and 10^{-5}

These values differ by at least ten orders of magnitude from the global minimum and clearly do not approximate it well.

Distribution of RMS errors for exemplars (DSM = 1)

There are two ways of counting and analyzing the distribution of RMS errors:

- Enumerate results for each random start, subsequent training run and refinement (method used thus far)
- Enumerate results for each local minimum, as represented by an exemplar (method used in this section)

Figure 7 shows the distribution of exemplar errors.

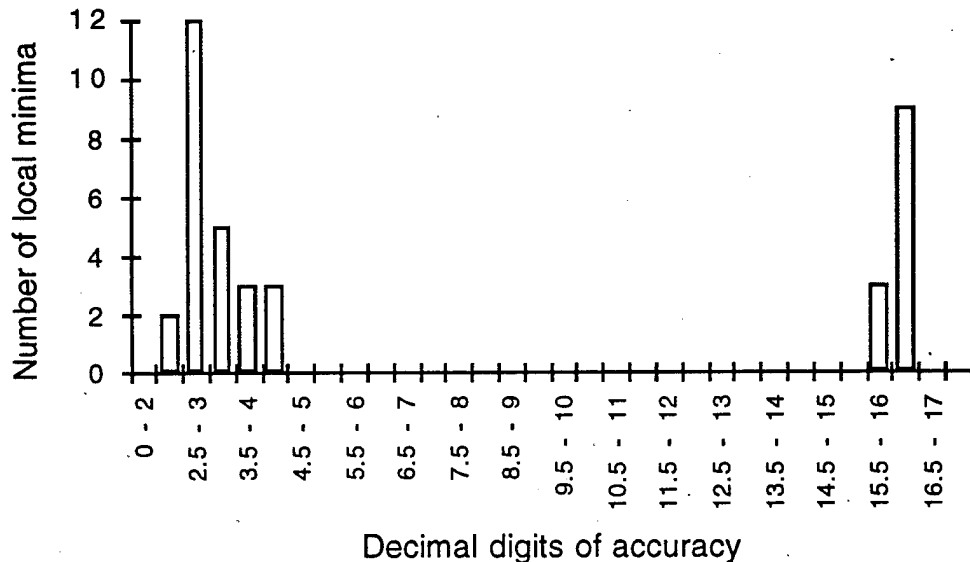


Figure 7: Distribution of RMS errors for exemplars (DSM = 1)

For DSM = 1, the distribution of RMS errors for exemplars differs significantly from the distribution for all solutions that refine (Compare Figures 6 and 7). The next table summarizes the differences numerically.

	Distribution of RMS errors	
	error > 10^{-5}	error < 10^{-15}
local minima	68% (25/37)	32% (12/37)
all refined solutions	12% (1027/8802)	88% (7775/8802)

The mean volume of the region of attraction for a local minimum with RMS error < 10^{-15} is over 15 times that for a minimum with error > 10^{-5} . The factor, 15, is obtained by computing the mean volume (number of solutions / number of minima) for each error range, and then taking the ratio of the means.

Fefferman [Fefferman, 1994] proved that, subject to a few technical conditions, two neural networks that have the same input-output mapping are isomorphic, i.e., they can be reduced to the same canonical form as discussed earlier in this paper. Figure 7 shows that 12 non-isomorphic networks (recall that local minima found in our studies are well-separated

after being reduced to canonical form) closely approximate the global minimum. Our results here are not counterexamples to Fefferman's uniqueness theorem. The theorem assumes complete knowledge of the input-output map, and, for the $DSM = 1$ case here, we only have 16 input-output pairs. A summary of Fefferman's proof and a presentation from the 1993 Neural Information Processing Systems conference (NIPS 93) are included in [Fefferman and Markel, 1994].

Classifying all candidate solutions

For each candidate solution, the refinement process ends in one of four ways:

- Refinement converged: refinement terminated successfully within 50 iterations; the candidate solution is a local minimum
- Refinement limit exceeded: refinement stopped after 50 iterations; the solution was still decreasing, but very slowly
- Hessian not positive definite: refinement stopped because at least one eigenvalue of the Hessian matrix was negative
- Eigenvalues are inaccurate: refinement stopped because the Hessian eigenvalues could not be computed accurately with the machine precision available (≈ 16 decimal digits)

Probability of refinement termination

Figure 8 shows, for each DSM, the probabilities of refinement ending in each of these 4 states. Note that zero probability outcomes appear as blank columns. For example, for the $DSM = 1/2$ experiment, all 20K refinements ended with the Hessian not positive definite.

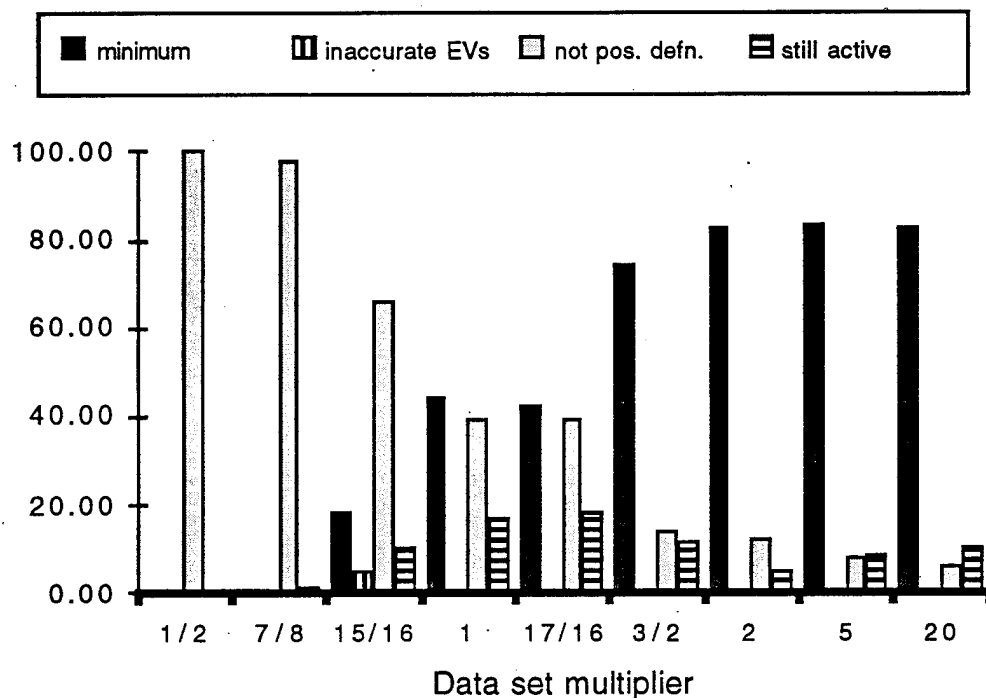


Figure 8: Probability that refinement terminates in each of four states

The following observations are apparent from Figure 8.

- high probability of finding a minimum for DSMs $\gg 1$
- low probability of finding a minimum for DSMs $\ll 1$
- high probability of terminating with a negative eigenvalue for DSMs < 1
- inaccurate eigenvalues occur only for DSMs < 1

The following table contains the basic data used to prepare Figure 8 and a column showing the number of local minima.

Termination status for all solutions					
		Number of solutions			
Data set multiplier	Number of local minima	Refined	Inaccurate eigenvalues	Not positive definite	Still active
1/2	0	0	0	20000	0
7/8	3	12	105	19536	347
15/16	195	3665	1061	13190	2084
1	37	8802	0	7825	3373
17/16	30	8470	0	7929	3601
3/2	11	14863	0	2793	2344
2	9	16529	0	2416	1055
5	4	16655	0	1637	1708
20	2	16572	0	1310	2118

Distribution of RMS errors for all solutions

Figure 9 includes 4 charts, one for each of the refinement termination states defined above. Each chart includes results for $DSM = 7/8, 15/16, 1,$ and 20 , which are sufficient to describe how the RMS errors depend on the training set size.

RMS errors are aggregated in three intervals whose choice was motivated by the results shown in Figures 6 and 7. There, RMS errors for local minima lie in one of two distinct regions, either less than 5 digits or greater than 15 digits of accuracy.

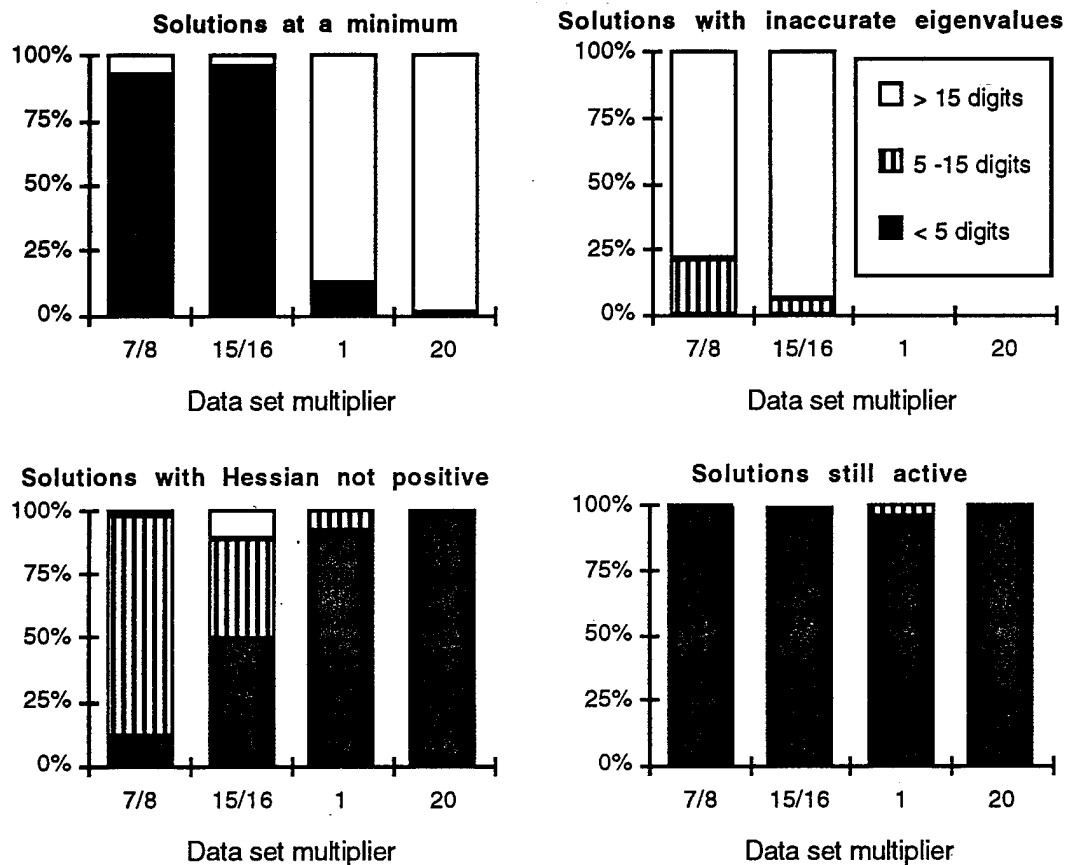


Figure 9: Distribution of RMS errors for all solutions

- Inaccurate eigenvalues occur only when attempting to refine solutions that closely approximate the global minimum
- Few solutions whose refinement terminates with the Hessian not positive definite closely approximate the global minimum
- Most solutions whose refinement terminates upon reaching the upper limit on Newton iterations have RMS errors at least 10 orders of magnitude greater than the global minimum
- Figures 8 and 9 result from data shown in the table below.

Distribution of RMS errors for all solutions				
		Number of solutions		
Termination condition	DSM	error > 10^{-5}	$10^{-5} > \text{error} > 10^{-15}$	$10^{-15} > \text{error}$
Minimum	7/8	11	0	1
	15/16	3478	0	187
	1	1027	0	7775
	20	93	0	16479
Inaccurate eigenvalues	7/8	0	22	83
	15/16	0	59	1002
	1	0	0	0
	20	0	0	0
Not positive definite	7/8	2086	17058	392
	15/16	6480	5183	1527
	1	7173	652	0
	20	1310	0	0
Still active	7/8	345	2	0
	15/16	2083	1	0
	1	3212	161	0
	20	2118	0	0

Distribution of RMS errors for exemplars

Figure 10, when compared with Figure 9a, shows how the distribution of RMS errors for all refined solutions differs from that for local minima, i.e., exemplars.

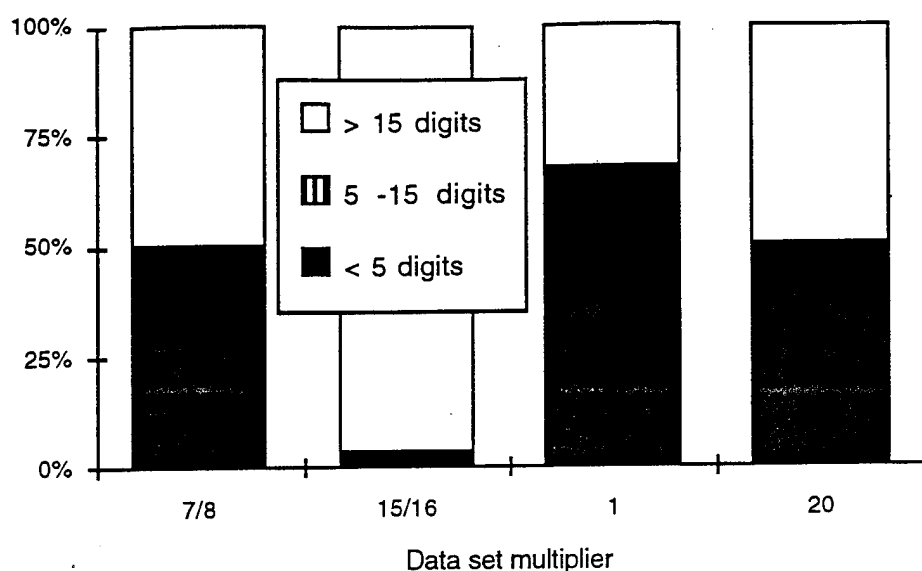


Figure 10: Distribution of RMS errors for exemplars

Note especially the striking difference in results for DSM = 15/16. Here, 187 SQP solutions refined successfully to 187 local minima, each of which closely approximates the global minimum. The other 3478 solutions that refined successfully all converged to the 7 local minima with RMS errors $> 10^{-5}$.

Data for Figure 10 are included in the following table.

Distribution of RMS errors for exemplars				
		Number of exemplars		
Termination condition	DSM	error $> 10^{-5}$	$10^{-5} > \text{error} > 10^{-15}$	$10^{-15} > \text{error}$
Minimum	7/8	1	0	1
	15/16	7	0	187
	1	25	0	12
	20	1	0	1

ERROR SURFACE DEPENDENCE ON NETWORK COMPLEXITY

Experiments run for 3-N-1 networks

For the results in the previous section, we fixed the network topology and varied the DSM. Here, we fix the DSM = 1, vary the topology, and use 10 targets instead of 1. The following table shows the number of data points (and weights since DSM = 1) for 3-N-1 networks. 20K random starts and 10 targets were used for each N, a total of 1 million training runs.

N	1	2	3	4	5
# data points	6	11	16	21	26

Relation between number of minima and number of random starts (DSM = 1)

First, we show, for each N and one target, how the number of minima depends on the number of random starts. (Figure 11)

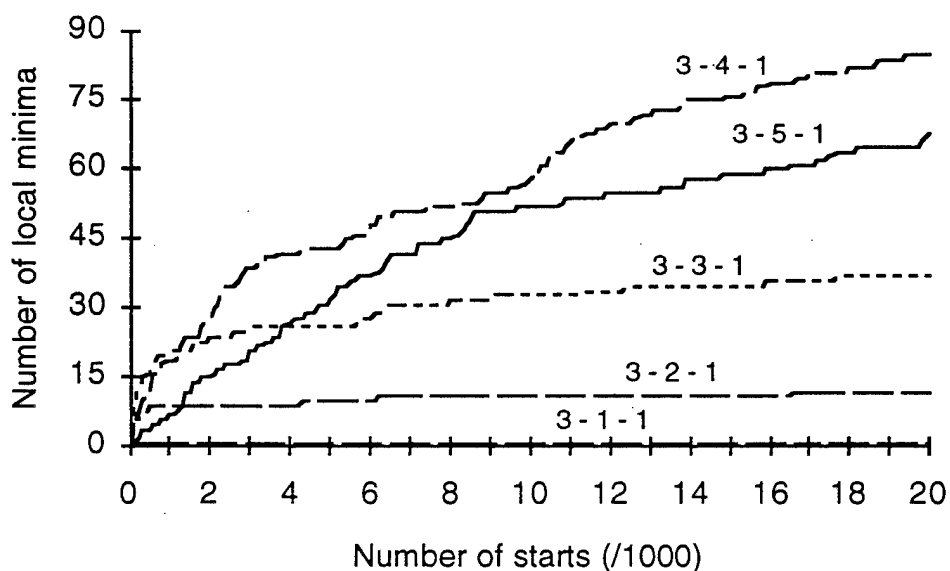


Figure 11: Minima vs. number of starts (1 target)

The lower 3 curves, for N = 1,2,3, approach asymptotes. Thus, 20K starts appear to be sufficient to obtain reliable estimates of minima. Note that the

curve for 3-5-1 actually lies below the curve for 3-4-1 and that it has not yet begun to flatten out. Clearly, 20K starts are not sufficient to obtain reliable counts of minima for 3-5-1 networks. Three days on a Sparc 10 were required to run the 3-5-1 network with 20K starts and 1 target. Since more (perhaps 100K) starts could be required, and we wanted to use 10 targets in order to estimate variance, we chose not to gather complete data for 3-5-1 networks. We discuss the 3-4-1 case further below.

Figure 12 shows for $N = 1, 2, 3, 4$, and 10 targets, how the mean number of minima depends on the number of random starts.

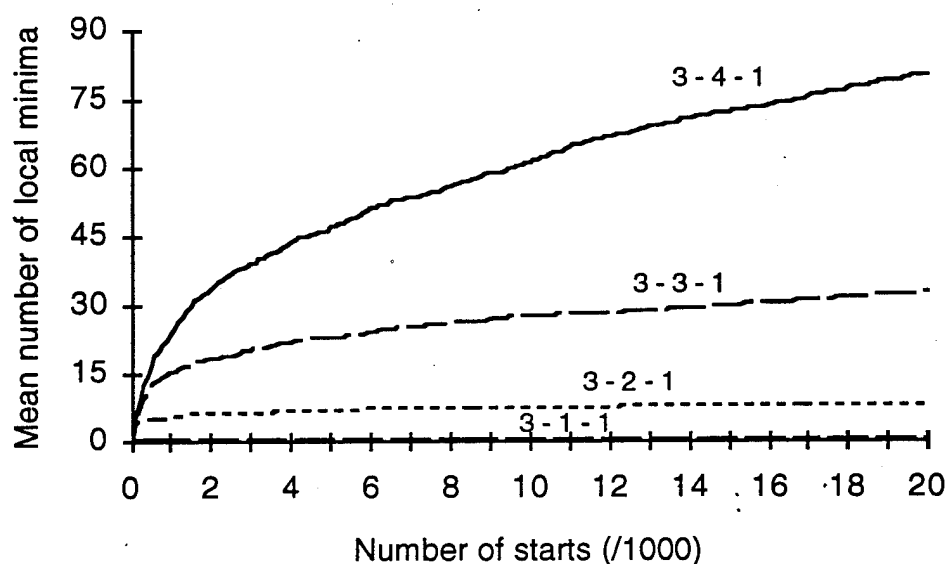


Figure 12: Minima vs. number of starts (10 targets)

Averaging results obtained with 10 targets clearly produces smoother curves. The plots for $N = 1, 2$ are near or at their limiting values. The plots for $N = 3, 4$ are still increasing gradually. Thus, the actual number of local minima for 3-3-1 and 3-4-1 networks is underestimated by using only 20K starts.

We estimated asymptotic limits for the number of local minima for 3-3-1 and 3-4-1 networks by using a simple asymptotic expression,

$$y = \frac{a}{1 + \frac{b}{x^2}},$$

where a and b were calculated from the data in the following table. The asymptotes, i.e., values of a, for 3-3-1 and 3-4-1 networks are 40 and 96, respectively.

	Mean number of local minima			
# of starts	3-1-1	3-2-1	3-3-1	3-4-1
19K	1.1	8.4	32.5	79.1
20K	1.1	8.4	33.1	80.5

Figure 13 shows, for 3-4-1 networks, the match between the experimental results with 20K starts and the asymptotic extension. 3-3-1 results are similar.

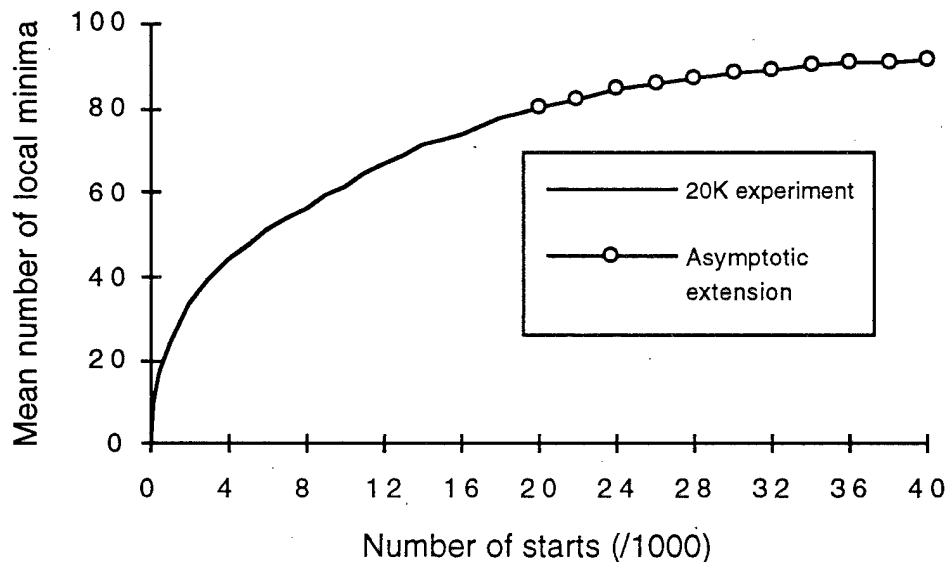


Figure 13: Minima vs. number of starts (3-4-1 network)

This approximation is good enough to justify its use to estimate the number of local minima that would result from running larger experiments. Use of only 20K starts appears to cause the number of minima to be underestimated by less than 18% in both cases.

Increase in minima with network complexity

Figure 14 corresponds to Figure 3 in the Introduction, but includes a variance estimate, based on 10 targets. As described above, the numbers of

local minima for 3-3-1 and 3-4-1 shown here appear to be underestimated by about 18%.

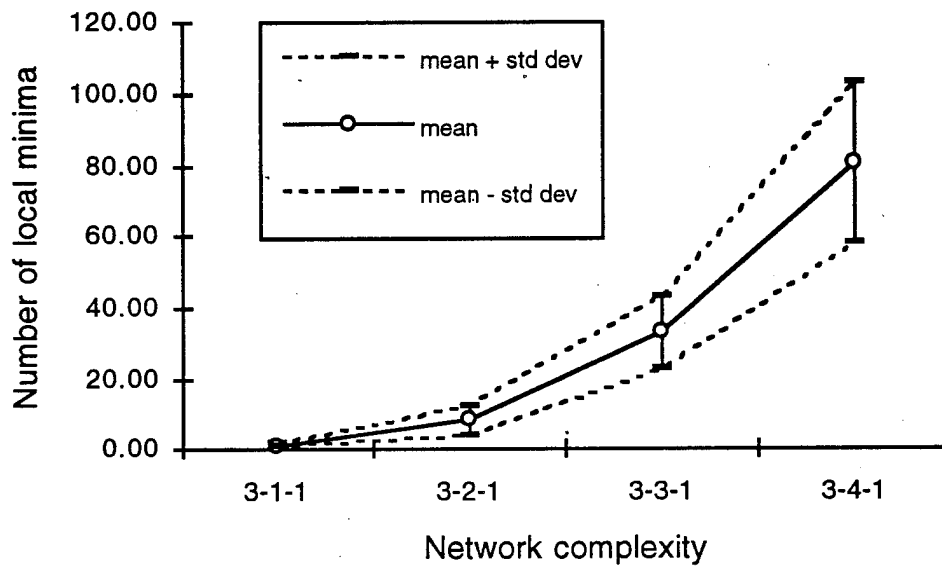


Figure 14: Minima vs. network complexity (10 targets)

Clearly, the number of minima found by experiment differs when different targets are used. This is to be expected since different subsets of the multi-dimensional weight space (up to 21 dimensions in Figure 14) are sampled when different targets and random starting points are used.

CONCLUSIONS

Error surface dependence on the Data Set Multiplier

We have shown, for a fixed topology, that the number of local minima is a decreasing function of the Data Set Multiplier (DSM). The probability that a random start will refine to a minimum also depends on the DSM. We next summarize DSM dependence in three regions: $\text{DSM} < 1$, $\text{DSM} \approx 1$, and $\text{DSM} \gg 1$. DSMs for the relevant experiments are shown in parentheses.

DSM < 1 (1/2, 7/8)

The probability of finding a minimum for DSMs in this range is very low. Refinement nearly always terminates with a very small gradient and at least one negative Hessian eigenvalue. The error surface is likely very complex. Although an infinity of "exact" solutions are possible with more parameters than data points, relatively few solutions with RMS errors $< 10^{-15}$ result from the refinement process.

DSM \approx 1 (15/16, 1, 17/16)

The largest numbers of local minima were found for DSMs ≈ 1 , e.g., 37 minima for $\text{DSM} = 1$, and 194 for $\text{DSM} = 15/16$. The distributions of RMS errors vary substantially for $\text{DSM} \approx 1$. Such variation is not surprising, given the prediction from linear theory described above.

DSM \gg 1 (5, 20)

There appear to be relatively few minima for DSMs in this range, and a high probability that refinement will converge to a minimum. Almost all the refined solutions had RMS errors $< 10^{-15}$. All those that failed to refine successfully had RMS errors $> 10^{-5}$.

Minima and network complexity

As one would expect, the number of local minima increases as network complexity increases.

Experimental Methodology

To make accurate, reliable statements about the nature of the error surface and the number of local minima requires:

- careful experimental procedures and attention to detail
- high-quality numerical optimization software
- many training runs from randomly generated starts
- refinement and verification of solutions after optimization

Limitations of our approach

Depending upon the network architecture and the DSM, it may be impossible to use our methodology to count minima, since it may happen that few solutions refine successfully. Few solutions refined, e.g., when we used two or more hidden layers, or for 3-N-1 networks with $N > 5$.

Other experiments are possible with the same methodology. For example, significant number of solutions refine for M-3-1 networks for $M > 3$, although we did not do sufficient M-3-1 studies to include them in this paper.

Implications for solving application problems

When interpreting the results of training experiments, a designer of neural networks for practical applications should take into account whether the problem is data-rich or data-poor. Several, perhaps many, training runs are required to have reasonable confidence that a near-optimal solution has been found.

Future work

Our long-term goal is to develop highly-efficient, semi-automatic methods for designing neural networks based on sound mathematical principles. We regard the work presented here as an essential first step toward this goal. We used an excellent, proven optimization algorithm to study carefully a problem whose exact solution was known. In the future, we intend to:

- Perform comparable studies using batch and on-line back propagation instead of Sequential Quadratic Programming
- Augment the model problem to include a twice-differentiable regularization term and use the same experimental methodology to produce quantitative studies of generalization

- Use the same experimental methodology to train low complexity networks to approximate training data generated from a known neural network of high complexity
- Apply the results of our research to real applications

REFERENCES

- Bishop, C. (1992). Exact Calculation of the Hessian Matrix for the Multilayer Perceptron. *Neural Computation* Vol. 4, pp. 494-501.
- Buntine, W. L. and Weigend, A. S. (1993). Computing Second Derivatives in Feed-Forward Networks: a Review. *IEEE Transactions on Neural Networks*. Vol. 5, #3, pp. 480-488.
- Chen, A. M.; Lu, H. and Hecht-Nielsen, R. (1993). On the geometry of feedforward neural network error surfaces. *Neural Computation* Vol. 5, pp. 910-927.
- Fefferman, Charles (1994). Reconstructing a neural network from its output. *Revista Mathematica Iberoamericana*. Vol. 10, #3, pp. 507-555.
- Fefferman, C. and Markel, S. (1994). Recovering a feed-forward net from its output. *Advanced in Neural Information Processing System 6*, Morgan-Kaufmann, pp. 335-342.
- Hamey, Leonard (1995). Analysis of the error surface of the XOR network with two hidden nodes. Macquarie University (Australia) Computing Report 95/167C.
- Hertz, J; Krogh, A. and Palmer, R.G. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, CA.
- The Numerical Algorithms Group Inc. (1991). Algorithm E04UCF. *The NAG Fortran Library Manual*, Mark 15. Downers Grove, IL.

ACKNOWLEDGMENTS

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract F49620-92-C-0072. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

We appreciate Margaret Wright's help in interpreting our SQP results, particularly early in the project when the results differed from our expectations. We also acknowledge, with thanks, the support of our colleagues Patrick Hsieh, Mark Plutowski, Clay Spence, and Ronald Sverdlove. Their comments and suggestions offered freely in many discussions during the course of this work are greatly appreciated.

APPENDIX

Glossary

- Clustering: the process of aggregating refined solutions that are equivalent
- Data set multiplier (DSM): a ratio; the number of input-output pairs in the training set / number of network weights
- Decimal digits of accuracy: N digits implies that the mean square error is approximately 10^{-N}
- Eigenvalues are inaccurate: refinement stopped because the Hessian eigenvalues could not be computed accurately with the machine precision available
- Exemplar: an arbitrarily chosen representative from an equivalent set of refined solutions, i.e., a cluster
- Hessian not positive definite: refinement stopped because at least one eigenvalue of the Hessian matrix was negative
- Input-output pair: one data point to be fit in a training problem; a vector of network input values and a single, desired network output value
- Local minimum: an exemplar
- Random start: a set of randomly generated network weights defining the point used to start a training experiment; random starts and targets are generated by the same algorithm
- Refinement: the iterative process used to pass from approximate to highly accurate local minima
- Refinement converged: refinement terminated successfully within 50 iterations; the candidate solution is a local minimum
- Refinement limit exceeded: refinement stopped after 50 iterations; the solution was still decreasing, but very slowly
- Refinement parameter: refinement converged when successive iterates of the network weights changed by less than this number; typically 10^{-10}
- SQP: Sequential Quadratic Programming; an optimization algorithm for minimizing a nonlinear function subject to bounds and linear and nonlinear constraints

- Target: a set of randomly generated network weights used to define training data; random starts and targets are generated by the same algorithm
- Training experiment: an SQP optimization run followed by refinement

**AUTOMATING BREAST CANCER DETECTION
BY
NEURAL NETWORK CELL ANALYSIS**

PART I: PROJECT DESCRIPTION AND ANALYSIS OF NEEDS AND REQUIREMENTS

Mark E. Plutowski, Ph.D.

David Sarnoff Research Center
Princeton, New Jersey, USA.
mplutowski@sarnoff.com

ABSTRACT

This report describes preliminary results of applying neural network classification models to automate breast cancer detection from cell image data.

Pathologists need computer aids to more quickly and reliably classify cells extracted from tissue suspected of being cancerous. However, this need conflicts with the requirement that such tools be reliable, easy to use, and able to be efficiently incorporated into current medical practice. This project explored the potential for automating cell classification using neural networks, and for incorporating these techniques into clinical application. All the results reported here were obtained on clinical breast cancer data.

These results are exploratory, due to the limited time frame of this project (one person at half-time over a period of 3.5 months).

OVERVIEW

Part I describes the problem domain, needs and requirements.

Part II describes the analysis that was performed on the breast cancer data.

Part III gives the results and final conclusions.

SUMMARY

This report describes the progress made in exploring the potential benefits of applying neural networks to automate breast cancer detection.

The figure below illustrates the current methodology for diagnosing a medical tissue sample, and points out the potential for additional automation. Given a tissue biopsy, a tissue sample is prepared for analysis. The sample is comprised of cells which can be categorized into a number of classes (illustrated by the box on the left). Under the currently most common methodology, this is viewed under a microscope by a medically trained expert or physician, who then provides a diagnosis as to the overall makeup of the tissue sample. This is illustrated in the figure by the topmost pathway from the cell image to the diagnosis end goal. This procedure relies primarily upon subjective evaluation, and results in a diagnosis that is essentially qualitative (say, in terms of a positive or negative classification of the entire sample as a whole).

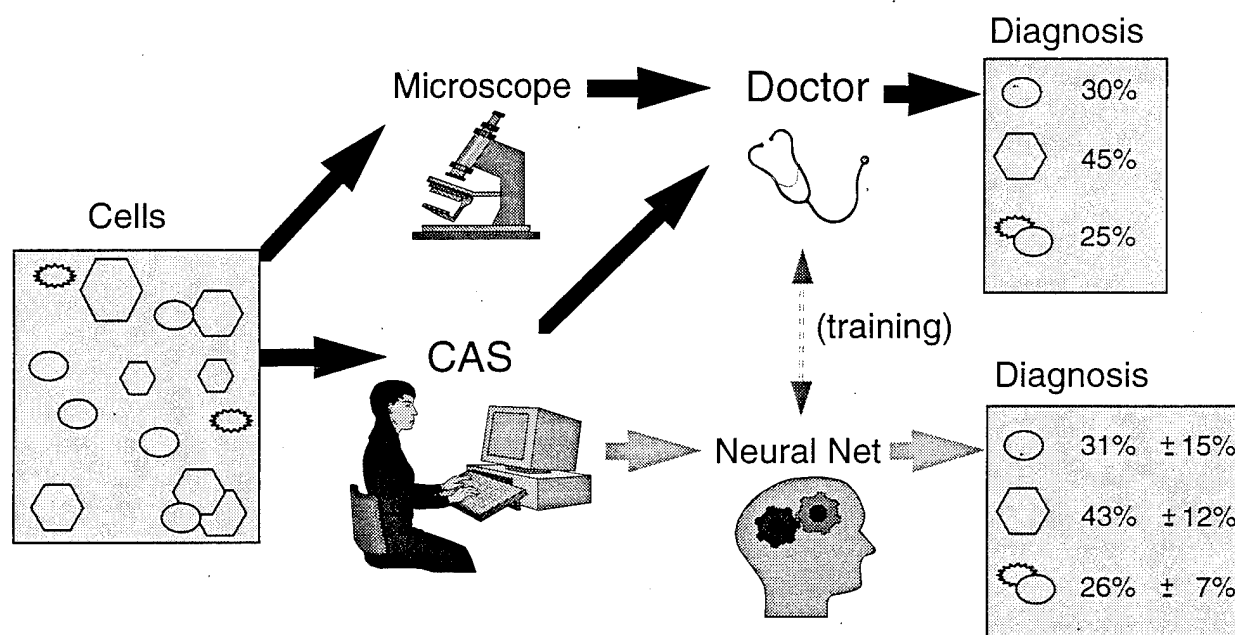


Figure 1. Methodologies for tissue sample analysis

With the aid of computerized tools that are currently available this procedure can be made more precise. These computer aids allow the human expert to evaluate each cell individually to provide a diagnosis of the given sample in terms of the class makeup. This is illustrated in the figure by the middle pathway from the cell image to the diagnosis end goal. While this methodology has many important and useful advantages over the conventional

method, it has the drawback of being time consuming and tedious to perform. This contributes to the fact that these computer-aids are not in widespread use.

Fortunately, potential exists to automate major components of this computerized method by using neural network classification methods. Furthermore, the results reported here were obtained by using tools and methods that can be incorporated into current clinical practice.

INTRODUCTION

After the initial detection of potential tumors by clinical evaluation (including mammography), cells are extracted from the patient in order to obtain more definitive information about the tumor. The extracted cells are then classified by a pathologist as malignant, normal, or other types. This is currently the most common means of analyzing a tissue sample. In Figure 1, this methodology is represented by the arrows along the path "Cells-Microscope-Doctor-Diagnosis."

Statistics on the properties of the cells from a tumor provide information about the aggressiveness of the tumor as well as other characteristics, such as the likelihood of disease-free survival if the tumor is removed, or what treatment might be best-suited to the case. The classification and statistics-gathering procedures are quite tedious and suffer from the necessarily subjective nature of the pathologists analysis.

Cell image analysis systems are available (e.g., Becton-Dickinson's "Cell Analysis System" line of workstations, in particular, the CAS 200, which was used for this study) which automate some of the tasks involved, and which furthermore allow the doctor to be more precise in evaluating the sample. It allows the user to classify the cells individually, thereby obtaining a more precise diagnosis, while simultaneously utilizing the microscope view to provide additional information that might not be available at the cell level. The CAS workstation therefore allows the doctor to provide a more precise analysis. In Figure 1, the path "Cells-CAS-Doctor-Diagnosis" illustrates how these computer aids are used to supplement the current methodology.

These computer-aids are also capable of providing rudimentary filters for classifying cells automatically. Currently, these filters are inherently inaccurate. In practice, the pathologist needs to correct and review each of the decisions made by the autoclassification filters. However, it turns out that the CAS workstation used in this study provides additional statistics on each individual cell that are currently not utilized by the CAS autoclassification filters. Therefore, the CAS workstation provides additional functionality which is currently not used by the doctor.

Using a commercially available software package (JVB Imaging's "Cell Sheet for Windows" program) these statistics can be extracted from the CAS output and put into a standard and easily useable format. The availability of

these additional statistics creates the possibility for further automation of the diagnostic procedure by machine-learning methods (such as neural networks).

These statistics can be used to train a neural network to estimate the doctor's performance. At present, the performance of a medical practitioner on a particular case is subjective, may vary from case to case, and, without aid of the CAS workstation, is necessarily either very tedious or imprecise. However, a neural network appropriately trained over a large number of cases may be objective, quantifiable, and reproducible, hence providing the benefits due to computer aids while eliminating much of the additional work required by current computer aids. Furthermore, this approach can make the experience of one or more human experts widely available, by drawing upon the experience of human expertise acquired by training.

The approach studied in the remainder of this report is illustrated in Figure 1 by the pathway Cells-CAS-NeuralNet-Diagnosis. Note that the diagnosis provided by the neural network in this figure contains an additional column. This corresponds to a "certainty" associated with the neural network's estimate. This "certainty factor" is an important component of the approach studied in this report, and is discussed in more detail below.

These machine-learning methods can be incorporated directly into the CAS workstation, extending its functionality while improving upon its ease of use.

Scope of applicability:

Similar techniques are applicable to a variety of diagnostic areas, e.g., bladder cancer (and other genitourinary cancers), prostate cancer, colon cancer, esophageal cancer, brain cancer, head and neck cancers, and several types of skin cancers, to name just a few of the most important examples. Therefore, results described here have a wide scope of application to essentially any other kind of cell sample which can be analyzed by cell image analysis workstations such as the CAS.

BACKGROUND

RESEARCH OBJECTIVES

The overall objective is to transfer neural network technology and expertise to the commercial medical world. In particular, the research goal is to develop pattern classification algorithms that greatly improve current cell sorting and analysis algorithms, so that the pathologists' work will be made more efficient, reliable, and accurate. Therefore, along with the usual analysis of the scientific issues involved with applying neural network learning methods to this data set, this report also includes an analysis of the issues involved in incorporating these techniques into clinical practice.

MEDICAL ISSUES

According to Dr. Siderits, several medical research articles have recently stated that subjective nuclear grading should be performed because it is

- 1 . related to survival,
- 2 . largely independent of the kind of tumor in a given tissue, and,
- 3 . is simple.

However, while nuclear grading has prognostic import, it is currently a qualitative procedure. One cannot expect mathematical precision from such a subjective approach. Automating this procedure would provide the medical benefits listed above, while dramatically improving the reliability and consistency of the diagnoses. However, it is equally important to ensure that these methods can be easily incorporated into clinical application.

BENEFITS, NEEDS, AND REQUIREMENTS

This section elaborates upon the needs, requirements, and potential benefits of this research.

Needs and Requirements

Patients need to have the large amounts of data now available via computerized methods made available to their care-providers.

Pathologists need computer aids to more quickly and reliably classify cells extracted from tissue suspected of being cancerous. However, this need conflicts with the requirement that such tools be reliable, easy to use, and able to be efficiently incorporated into current medical practice. This includes the requirement that the analysis provided be meaningful and easy to interpret.

Manufacturers of cell image analysis workstations need to make their workstations easier to use, in order to make them more appealing to medical institutions.

Benefits to the Care-receiver

More objective and accurate diagnosis (via improvements in cell classification) will improve the pathologist's prognosis, and hence, the relevance of the disease treatment. This may, for example, reduce unnecessary surgery. Patients are also impacted indirectly by making the process of cell classification more efficient, thereby either decreasing the amount of time required by critical human resources for performing tedious image processing (thereby reducing the cost of medical care), and by allowing much more data to be processed in the same amount of

time (thereby improving the diagnostic capability of the medical staff for a difficult case). They impact all health care users as well, even if they never become patients themselves, by potentially reducing the cost of cancer detection.

Benefits to the Care-provider

Better use of the available information provided by computer-based cell-imaging workstations should allow us to better automate two of the most important aspects of cell classification, namely, culling out superfluous data, and automatically classifying the remaining data.

These benefits impact the medical staff and hospital management directly in terms of time saved, now required by critical human resources for processing cell image data that is now available due to systems such as the CAS workstation and the Cell Sheet software package.

Furthermore, although this is a subject for continued investigation which we were not able to evaluate directly in the short amount of time we spent on this project, we surmise that with the appropriate modification of the methods we studied here, it would be possible to provide the user with the capability to easily recalibrate the cell classification filters. This capability would be most important for training the system to recognize new or rare sets of classes.

Clinical Application

An important goal of this project will be to incorporate our results into the clinical workplace. This serves two purposes:

- 1 . Validate empirical results in clinical use.
- 2 . Validate ease-of-use in clinical use.

Preliminary results done to date on this data (for the ARPA contract) demonstrate that a properly trained neural network model can automate important components of the doctor's diagnosis (by using information which is effectively unavailable to the doctor's eyes). Despite these encouraging technical results, the task of introducing this technology into the clinical setting may be difficult to achieve. This potential difficulty may be alleviated by two key findings.

The first finding regards a technical aspect of the approach we are using. The preliminary results for the ARPA contract demonstrate that a properly trained neural network model can not only automate components of the doctor's diagnosis, but may also be able to compute a "certainty factor" for the diagnosis. In the illustration above, note that certainty factors are associated with the class probabilities given by the neural network. This allows the neural

network to flag difficult cases for review, thereby automatically notifying the medical expert for those cases where additional expertise is needed to resolve ambiguous results.

The second important finding regards a characteristic of the working clinical environment. There is currently a billing code (a.k.a. CPT code) for "morphometric analysis of tumor tissue." This means that doctors can be reimbursed for utilizing image analysis platforms. The task of introducing this methodology into the hospital workplace is made easier by the fact that structure is already in place to allow doctors to be reimbursed for this advanced technology by current administrative procedures.

Additional work will be done to evaluate methods for improving the success of incorporating these methods into the current clinical environment. If the certainty factors prove successful in clinical application, then it is possible to use them to combine several neural network models. This serves two purposes:

- 1 . to evaluate whether it is possible to improve the reliability of the neural network models by diversifying risk over a set of trained models, and
- 2 . to evaluate the possibility of combining the neural network models with other methods for which certainty factors are available.

PARTICIPANTS, COLLABORATORS, AND TOOLS

This section lists the parties that were involved in this study.

- 1 . Helene Fuld Medical Center, Department of Experimental Pathology, Trenton, New Jersey.
- 2 . Richard Siderits, M.D. Head of Experimental Pathology, Helene Fuld Medical Center, anatomic and clinical pathologist, and member of the Diplomate College of American Pathologists, Board-certified in Anatomic as well as Clinical Pathology, on the Diagnostic Immunology Resource Committee of the Council of American Pathologists, as well as on its Image Analysis sub-committee.
- 3 . DeeAnn Wolf, Research Team Coordinator, Digital Bioanalysis Department, Experimental Pathology, Helene Fuld Medical Center. Ms. Wolf is trained as a medical technician. She was the principal technical expert in Dr. Siderits' laboratory regarding the interface between biological data and the computer-based cell image analysis workstation.
- 4 . Becton Dickinson Inc, a leading manufacturer of medical instrumentation hardware and tissue sample preparation materials. They manufacture the CAS cytology imaging system, used diagnostically for a variety of cancers. Workstations manufactured by Becton Dickinson comprise about 80% of the installed base, with about

500 of these machines sold worldwide. Becton Dickinson provided Dr. Siderits' with the CAS 200 system as part of a previous agreement with the Helene Fuld Medical Center.

5. JVB Imaging Inc, biomedical software developers, with expertise in advanced software tools for cancer cytology and computer aided diagnosis.

Dr. Siderits' lab is an R&D site for Becton Dickinson's Cell Analysis Systems, which means that they have a cell image analysis workstation and the expertise required to operate it.

The imaging workstation used in this project by Dr. Siderits' laboratory is the Becton Dickenson CAS 200. The price of this workstation is roughly \$180,000. This workstation was used to generate statistical data and cell images from cell samples. JVB Imaging's "Cell Sheet" program gave us the ability to easily process the data produced by the CAS 200, and most importantly, to easily transform it into a standard format.

GENERAL CHARACTERISTICS OF THE CELL DATA

Dr. Siderits generated training data which we used to train statistical models to estimate the probability that a cell belongs to a particular class of cells (e.g., "cancerous").

Here's some fundamental terminology. As this type of analysis looks at a particular cell individually (i.e., is performed at the cellular level) it is "cytology", as compared to "histology" (which is performed at the tissue level).

We defined the learning task by determining the type of tissue (here, breast tissue) and the types of cells in that tissue that will be of interest henceforth (6 main types, to be defined next). We then "class" the cell data by constructing cell classes, each class comprised of cells from 1 or more of the 6 main types (we elaborate upon this below in the section on Neural Network Training Methods). The following list describes the 6 main cell types. Each item in the list is headed by the name by which we refer to the type henceforth. Where appropriate, the heading name is followed by the scientific name, followed by a brief description in layman's terms, with some additional background.

1. Typical Normals

Scientific name: Normal duct epithelium.

Normal duct tissue cells.

Background: Epithelium is a general type of cell that acts as a surface lining. The function of epithelium can vary, from protecting a border (e.g., in plants, as bark, and in animals, as skin), to providing filtration (e.g., liver cells), to generating fluids (e.g., breast duct tissue, which are involved with producing milk).

2 . Atypical Normals

Scientific name: Atypical duct epithelial hyperplasia.

Layman's term: Normal duct tissue cells that appear to exhibit abnormal growth but which are noncancerous.

Background: Such duct cells are atypical (therefore "abnormal" in a particular sense) because they exhibit hyperplasia (abnormal growth). However, they are noncancerous, even though they have an abnormal appearance. The appearance can be abnormal for several reasons, including (a) cell damage, and (b) due to perturbations caused by the cell being in the vicinity of other cells, such as "inflammatory" cells (i.e., white blood cells) or cancer cells. Inflammatory cells and cancer cells tend to evoke abnormal responses from cells with which they come within close proximity.

3 . Benign

Scientific name: Neoplastic - benign.

Layman's term: Noncancerous but abnormal growth.

Background: Neoplasticity (literally, "new growth") is the state of growth exhibited by cancerous cells. A benign cancer cell can become tumorous, but does not spread rapidly, because a benign tumor spreads only by multiplying while remaining in the same general vicinity. In contrast, cancer cells that spread by "metastasis" do so rapidly by breaking loose from their ancestors and traveling far away and taking up root elsewhere, and cancer cells that spread by "invasive extension" do so by branching out into their neighboring vicinity. Benign cells are limited from these more rapid transport mechanisms because they are surrounded by scar tissue. This scar tissue protects the surrounding tissue and limits the growth of the benign cells.

4 . Cancer

Scientific name: Neoplastic - carcinoma (cancerous).

Background: This class is comprised of the nasty cancer cells that spread by metastasis or invasive growth (see also "Background" under Benign item above).

5 . Garbage and Nondiagnostic

Scientific name: Nondiagnostic.

Background: This class is comprised of cells that are either damaged, clumped together or overlapping (so that the automatic image segmentation fails to separate them into distinct cell bodies), as well as of parts of cells and other extraneous matter that is not small enough to be automatically filtered out but which is otherwise unclassifiable.

6 . White Blood Cells

Scientific name: Inflammatory cells (acute and chronic inflammatory cells).

Layman's term: White Blood Cells.

Background: The two types of white blood cells encountered in our data are the (a) neutrophils, and (b) lymphocytes. Neutrophils are also known as "acute" inflammatory cells, because these cells are the first on the scene, responding quickly to an invasion by foreign matter. Lymphocytes are also known as "chronic" inflammatory cells because they persist over time. In other words, these cells take longer to arrive, but hunker down in the trenches for the long haul. It is relevant to this study that images of these two types of cells appear to be much different.

In the following, we will refer to "Typical Normals" and "Atypical Normals" collectively as "Normals." In our main results, we lump Garbage cells together with the White Blood Cells to obtain what we refer to as the "Nondiagnostic" class. This class is comprised of all objects which are not useful for diagnostic purposes, because the objects do not correspond to a cell class of interest.

Therefore, while we conducted numerous exploratory experiments which tested our main results consider 4 main classes:

- 1 . Normal
- 2 . Benign
- 3 . Cancer
- 4 . Nondiagnostic.

INITIAL FEASIBILITY STUDY

This section documents the preliminary phase of this project, in which we worked along side medical research staff, examining their current working environment, and obtaining training data. We streamlined the data acquisition protocol, and performed a preliminary feasibility study of the potential benefits of this project.

Choices were available to us involving several issues:

- 1 . Methods for applying neural network training methods to improve upon the current application of the CAS workstation.
- 2 . Methods of collecting data from the clinical environment,
- 3 . Cell classes to use in this study
- 4 . A cost analysis (in terms of the amount of time required by the human expert to reclassify cells after an initial autotclassification by the CAS system) to help evaluate the need for additional automation of cell classification.

The results of this on-site analysis of the current clinical working environment were encouraging. To summarize, we expect to be able to improve the efficiency of two major aspects of the cell analysis and classification process. Furthermore, these improvements can be implemented without need for extensive retooling of the present working environment - these improvements could be implemented as extensions of a currently available software tool (or one like it), without need for additional hardware. Indeed, we expect it to be possible to make the current tools easier to use while providing them with additional capability.

To place our approach in the proper context, we now provide a simplified overview of the current situation. Becton-Dickinson is a corporation that manufactures a system (referred to as the Cell Analysis System, or CAS) which automates several major aspects of cell classification. This system is comprised (essentially) of a microscope, an imaging camera, a computer and related hardware, and a software program. A medical technician prepares a tissue sample for analysis (and calibrates the CAS equipment appropriately for the particular sample if necessary). Under the guidance of a medical doctor, the CAS imaging system analyzes the entire sample. This analysis locates each object in the sample, and computes several statistics for each object. Each object is passed through a crude filter, classifying the object into one of at most 6 classes. The classification is indicated by a visual aid (outlining an image of the object on a monitor with a certain color); this allows the human expert to quickly ascertain the initial classification, as determined by the CAS filters. Then, the human expert reclassifies each object in the sample (by pointing and clicking with a cursor tracking device, i.e., a mouse or trackball). At the same time, the human expert deletes any objects that do not correspond to cells with diagnostic potential (e.g., clumped cells, closely touching cells that could not be segmented by the CAS system, broken or damaged cells, extraneous matter, and overlays -

i.e., one cell occluding another). If data is at a premium, the expert or the medical technician may further improve on the CAS image segmentation by manually separating clumped groups of cells into their individual constituents.

They fine-tuned the CAS 200 "filters" to do a good first-cut approximation to the correct classification. Then, with the entire tissue sample in view under microscope, along with population statistics giving a histogram of the distribution of DNA among the cells, Dr. Siderits evaluated each case cell by cell, verifying the classification for each individual cell and changing it if necessary.

At present, performing cell classification (with the maximum amount of automation - i.e., without doing any additional work attempting to improve on the CAS image segmentation) requires 15 to 30 minutes for slides containing from 100 to 300 cells.

AREAS OF POTENTIAL BENEFIT

We recognized three areas for application of neural network training methods:

- 1 . Automate the detection of "garbage" cells.
- 2 . Improve upon the automatic cell classification.
- 3 . Provide a facility for automatically configuring the cell analysis filters given a small set of cells provided by the user.

The immediate goal of this project is to extend the existing system (either with an add-on software program, or by extending the CAS system itself) towards achieving the first two benefits.

Although outside the scope of the current project, the third item in the list above is an attractive option to the medical personnel whom we worked with on this project, because it provides a way of automating the currently tedious operation of "tweaking" the CAS filters to "fit" cell classes of interest.

SETTING UP THE DATA ACQUISITION PROTOCOL

Streamlining the data acquisition procedure was important to help lessen the impact of our study upon the clinical environment of the participants of this study. We acquired a few initial datasets, after which we settled upon a set of cell classes, along with a data acquisition procedure, that would provide us with the most informative dataset, while minimizing the effort required by the clinical staff.

In conjunction with Dr. Siderits and Ms. Wolf, we set up an initial data acquisition protocol, acquiring two data sets. Each data set contains two tables of data, the first giving the classification as performed by the CAS system (using filters calibrated and tweaked by expert personnel on prior data sets), and the second giving the classification

according to the human expert. Each table consists of one record for each cell, each record consisting of 37 fields, one field for the classification of the cell (the dependent variable), and 35 features (the independent variables). These 35 features provide information quantitatively characterizing a wide range of aspects of each cell, ranging from cell size and shape to the amount of DNA, as well as a variety of texture measures.

Recall that the current set of tools (the CAS workstation, along with the Cell Sheet software) generates 35 features for each object. Only 4 of these 35 features are used by the CAS filters for its autoclassification. A preliminary data analysis of the first two data sets (the first comprised of 131 cells, and the second, of 353 cells) indicated that several of the additional features are good indicators for cell classification. (We repeat this preliminary result in more detail below.) Note further that the CAS filters are limited to defining classes according to ranges on the feature parameters. Therefore, the user sets the filter by setting a minimum and maximum range for each of the 4 features; a cell is classified according to the first class for which it satisfies all of the range requirements. Therefore, we expect to be improve greatly upon the use of these 4 features as well, compared to the capability of the currently most advanced cell analysis system.

ADDITIONAL TASKS INVOLVED IN USING CURRENT COMPUTER AIDS

The following table shows our first attempt at selecting a set of cell classes. Because the CAS system currently only provides for 6 distinct classes, we settled upon classes corresponding to the six most important types of cells we expected to see in tissue samples.

Class	CAS 200 Filter		Human Expert	
	Number	%	Number	%
Typical-Normal	50	14.4	1	0.3
Atypical-Normal	80	23.1	68	21.2
Benign	0	0.0	0	0.0
Cancerous	198	57.1	202	62.9
Neutrophils	14	4.0	6	1.9
Lymphocytes	5	1.4	44	13.7
<hr/>				
Total:	347		321	

Note that the Human Expert classification includes fewer cells, as additional cells were deleted during the classification phase (these were less obvious "Garbage" that wasn't detected in the garbage collection phase that preceded the automatic classification by the CAS filters).

We realized that this set of classes did not account for the amount of "garbage" and "nondiagnostic" cells in the original tissue sample. Therefore, we renamed the "Neutrophil" class to "Nondiagnostic" and rather than having the nondiagnostic cells manually deleted from the sample, we instead had them put into the "Nondiagnostic" class, along with the neutrophils. We then tweaked the CAS filters to autoclassify this modified set of classes.

To illustrate the amount of human effort required to perform cell classification after filtering by the CAS system, here are the statistics for the next two tissue samples, with the "Filter" column representing the "before" stage, and the "Expert" column representing the results due to human classification and garbage detection.

Data Set 1			Data Set 2						
Class	Filter	Expert	Filter	Expert					
Typical-Normal		0	0.0%	0	0.0%	43	12.2%	0	0.0%
Atypical-Normal		7	5.3%	7	5.3%	74	21.0%	47	13.3%
Benign		0	0.0%	0	0.0%	0	0.0%	0	0.0%
Cancerous		120	91.6%	116	88.6%	218	61.8%	136	38.5%
Lymphocytes		0	0.0%	0	0.0%	0	0.0%	0	0.0%
Nondiagnostic		4	3.1%	7	5.3%	18	5.1%	91	25.8%

Totals: 131 cells 353 cells

Note the large discrepancy in the performance of this filter. In particular, note the work involved with respect to reclassifying the sixth class, which is comprised of "garbage" (extraneous debris, cell parts, clumped or overlapping cells) and cells that don't fall within the first 5 classes. Clearly there is room for improvement towards obtaining filters that work well across a wide variety of tissue samples. Also, note that the expert reclassifications for Data Set 1 were relatively few compared to Data Set 2. However, the amount of time required by the expert in processing Data Set 1 was not significantly reduced, as Data Set 1 required 17 minutes of the expert's constant attention, whereas Data Set 2 required 24 minutes.

These results indicated to us that there was room for improving the speed and accuracy of the CAS autoclassification, both in terms of correctly distinguishing between normal and cancer cells, as well as in flagging nondiagnostic objects.

CONCLUSIONS

Currently available computer aids provide additional useful functionality that is effectively unavailable otherwise. Be that as it may, their acceptance is not widespread, due in part to additional time and work required for their use. The potential benefits due to additional automation are significant; among these are

- 1 . Removing the current disincentives for using computer automation, hence removing barriers to their use.
- 2 . Paving the way for developing automation beyond simply making current technology easier to use - i.e., providing additional functionality that is currently unavailable.

Item 1 can be achieved by reducing the time and effort necessary to configure, utilize, and interpret the results of current computer aids.

Item 2 addresses an entire range of possibilities which are possible and desirable, including

- Complete automation of cell diagnosis.
- Using trained neural network models as "Expert Systems" thereby making expertise of select individuals available on a wider scale.
- Using internetworking to facilitate wide availability of expertise, either as a second opinion, as decision support to regions of the country where expertise is scarce, or as educational tools.

These benefits are theoretically possible, given sufficient amounts of training data, appropriate models, and appropriate training methods. However, in practice we work with finite amounts of data. Part II performs a data analysis to determine what aspects of the learning task appear to be learnable, and what aspects may either require substantially larger quantities of data or require a novel approach in order to deal with inadequacies in the available data. Part III presents the results of preliminary training runs, providing further corroboration of the challenges indicated in Part II, and culminating in the description of an approach that may meet these challenges while fulfilling the potential benefits described in the analysis of Part I.

AUTOMATING BREAST CANCER DETECTION

BY

NEURAL NETWORK CELL ANALYSIS

PART II: FEATURE ANALYSIS

II.1. OVERVIEW

Part II describes the analysis that was performed on the breast cancer data. We selected 11 out of the 35 available features. The first section provides a visualization of statistics used in the analysis. The next section describes the features in more detail. We conclude this chapter with an intuitive assessment of the challenges and opportunities presented by this learning task, as evidenced by this analysis.

II.2. DATA ANALYSIS: VISUALIZING FEATURE STATISTICS

The goal of the data analysis was to quickly select a subset of the features providing good indicators of the cell class.

For the convenience of the reader, we provide a brief description of the classes of interest.

- | | |
|-------------------|--|
| 1 . Normal | Includes both "Typical" normals and "Atypical" normals. |
| 2 . Benign | Benign tumor cells. |
| 3 . Cancer | Carcinoma. |
| 4 . NonDiagnostic | This class includes white blood cells (neutrophils and lymphocytes), clumped cells and cells that otherwise could not be automatically segmented by the cell image analysis workstation, as well as "garbage," such as damaged cells, bits of cell matter, and other debris. |

See Part I for more detail on these classes.

Although there exist several rigorous quantitative methods for performing such an analysis, due to the explorative nature of this project and the limited time available, we selected a subset of the features by visual analysis of statistics derived from the data. This provided us with a subset of 11 features out of 35 that were available. This approach also had the additional benefit of providing insight into the data.

To perform the visual analysis we viewed the conditional probability density function (PDF) of each feature given each of the 4 classes (Normal, Benign, Cancer, Nondiagnostic), as well as the conditional cumulative density function (CDF) of each feature given each class. We were able to quickly cluster the features according to similarity.

Recall that the Nondiagnostic class contains neutrophils and lymphocytes (white blood cells) as well as clumped and damaged cells which may be any combination of normal, benign, cancer, neutrophils and lymphocytes.

The PDFs and CDFs were created from a sample of 3319 cell images. The PDFs are estimated by histograms, using 150 bins for each histogram. The CDFs are graphed using line plots of the cumulative sums over the histograms used to obtain the PDFs. For simplicity, we binned the data in the range from zero to the maximum value of the feature of interest over all classes.

HOW THE PDFS AND CDFS WERE CALCULATED

Each of the following figures contains four conditional PDFs for a particular feature. Each conditional PDF gives the sample PDF of a particular feature given a particular class. We evaluate four classes: Normal, Benign, Cancer, and NonDiagnostic.

The same coordinates and dimensions are used for each set of four graphs to allow comparisons among the four classes. For each feature, the range of horizontal axis also contains information about the data. The maximum value

of the horizontal value corresponds to the value below which contains at least 99% of all the data. The minimum value of the horizontal axis corresponds to the first bin of any class that is nonzero, unless this value is less than 10% of the maximum value over all classes, in which case it is set to zero.

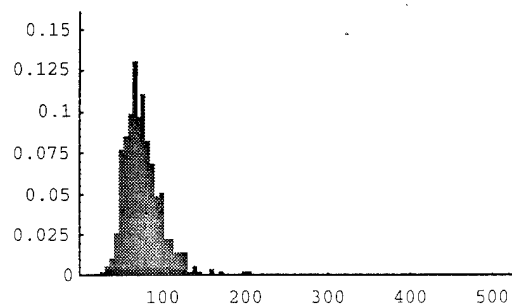
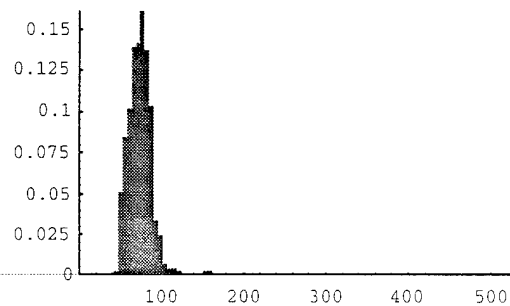
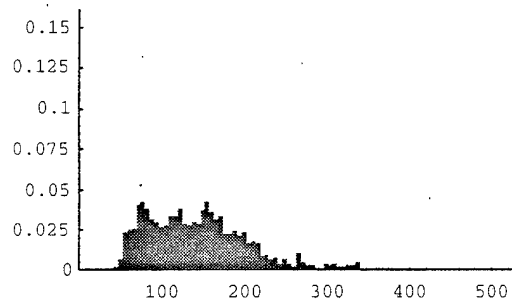
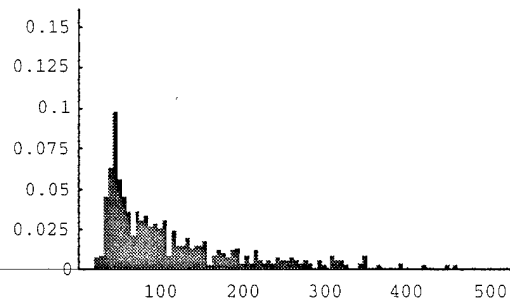
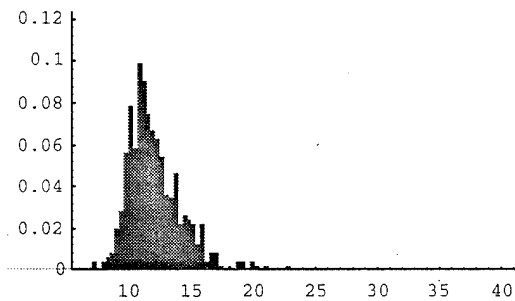
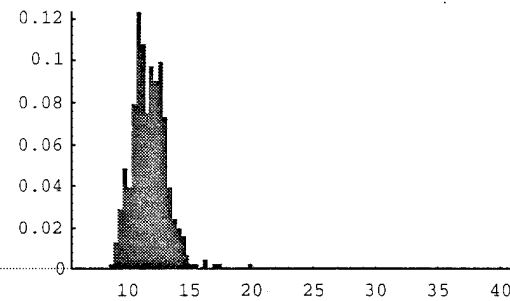
The binning procedure could be improved to better spread the data out among the 150 bins - the attentive reader will notice some artifacts of this binning procedure among the following figures. However, it served to provide an informative visualization of feature statistics, from which we were able to reduce the dimensionality of the feature space by 69 percent.

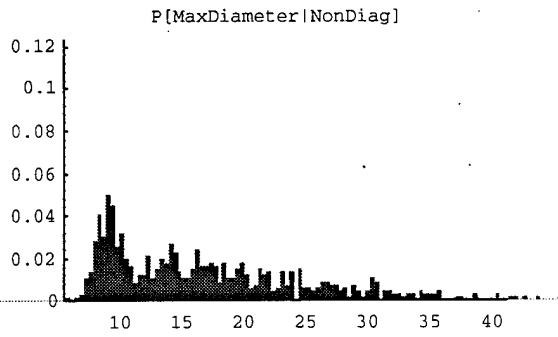
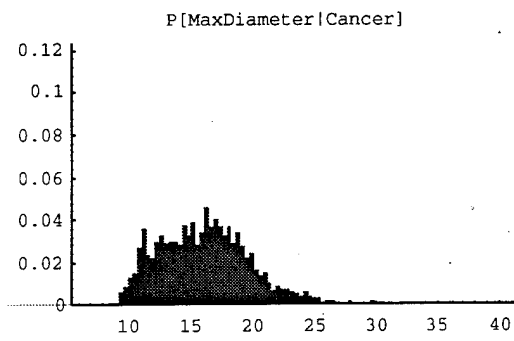
150 bins turned out to be sufficient for many features, overly fine-grained for some, and overly coarse for others. The choice of number of bins could also be improved, by choosing a number better suited for each feature individually; however, the sensitivity to number of bins is reduced by consideration of the conditional CDFs, which are shown following the PDFs.

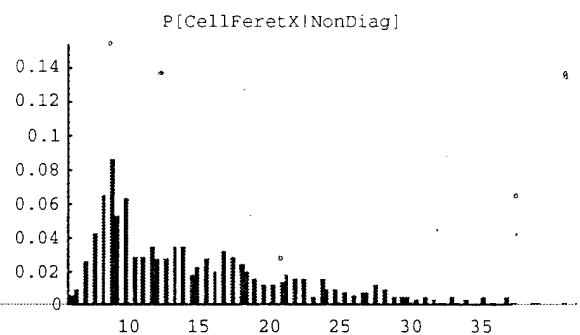
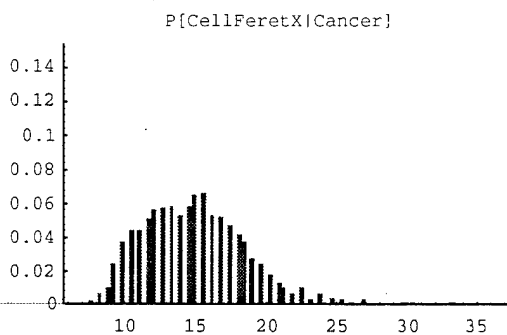
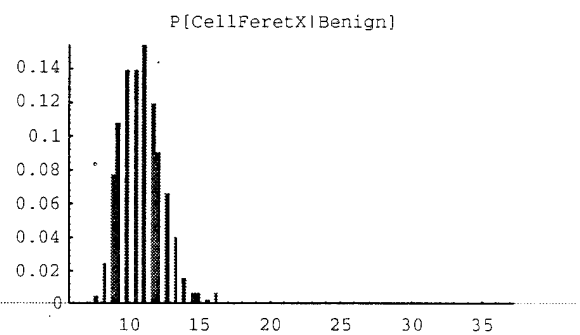
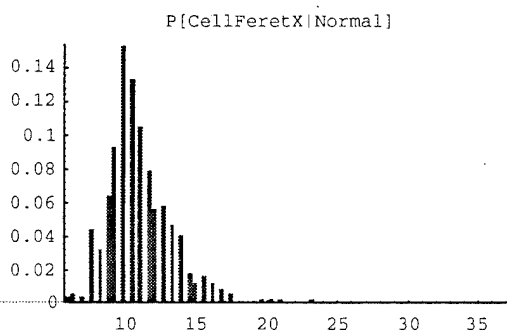
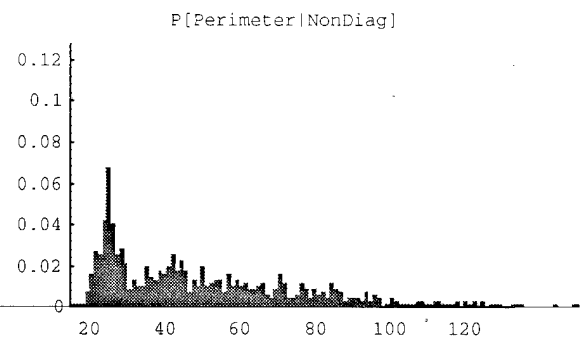
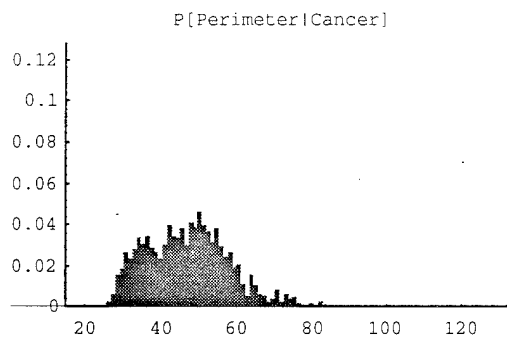
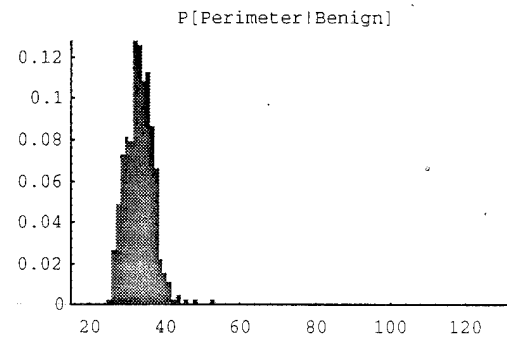
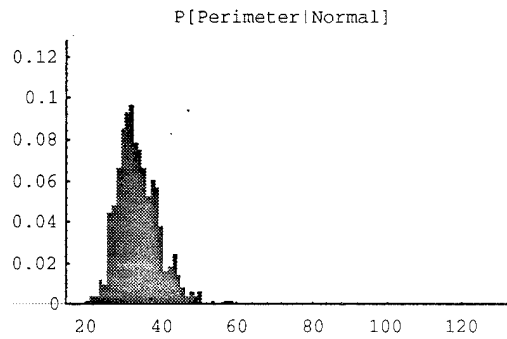
THE CONDITIONAL PDFS

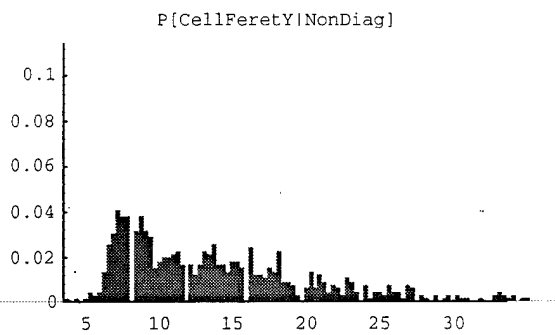
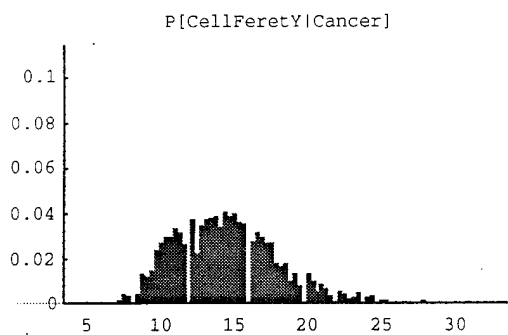
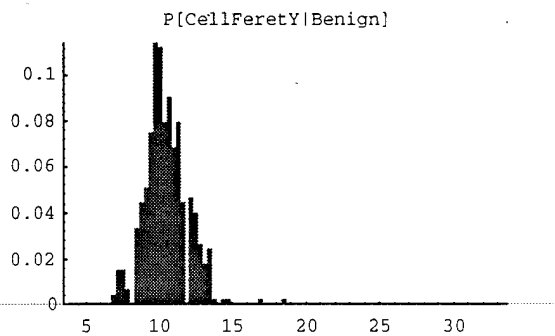
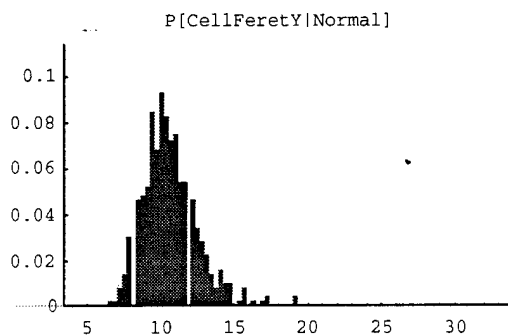
The conditional PDFs for each feature follow. The figures are grouped according to visual similarity. To see this, note general shape and location of the PDFs. Features measuring cell morphometry (e.g., cell size and shape) are shown first. Next are features which measure DNA content, then follow features which measure low order statistics about the image (based upon optical density, which is measured by gray scale intensity). The last 13 features measure different types of "texture."

The first group of features measure the size of the cell. Note that Cell Area is an especially good indicator of Cancer and NonDiagnostic cells. This is because Cancer cells tend to grow larger than either Normal or Benign, and because NonDiagnostic objects often correspond to clumped or overlapping cells.

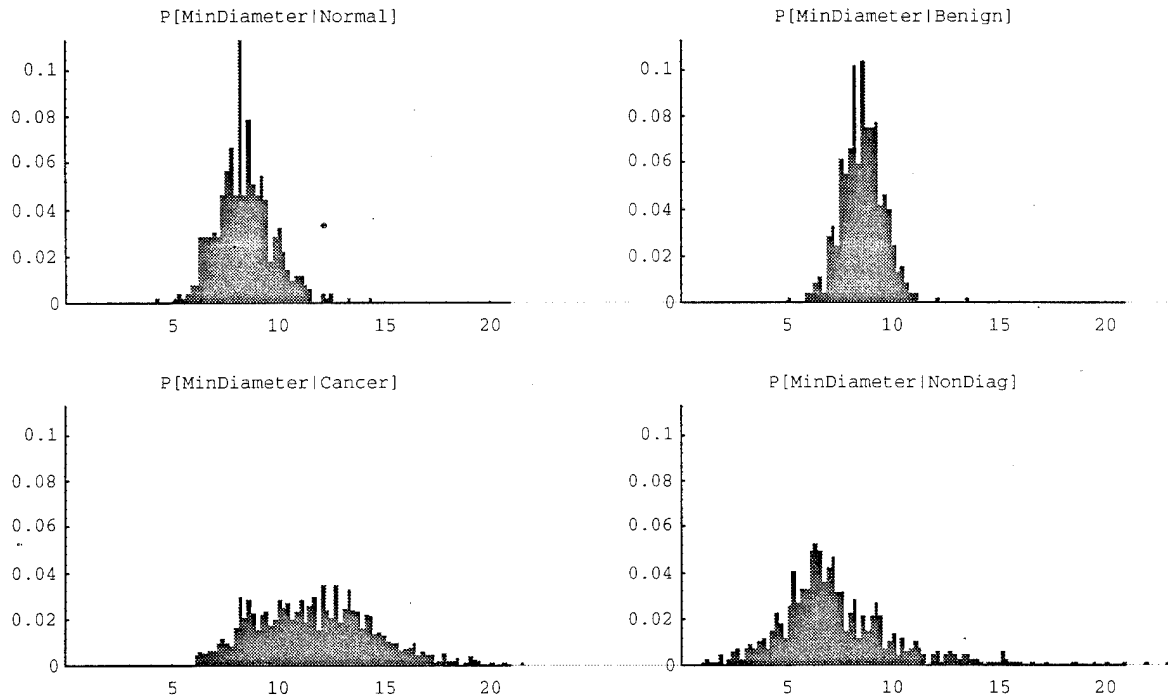
$P[\text{CellArea}|\text{Normal}]$  $P[\text{CellArea}|\text{Benign}]$  $P[\text{CellArea}|\text{Cancer}]$  $P[\text{CellArea}|\text{NonDiag}]$  $P[\text{MaxDiameter}|\text{Normal}]$  $P[\text{MaxDiameter}|\text{Benign}]$ 



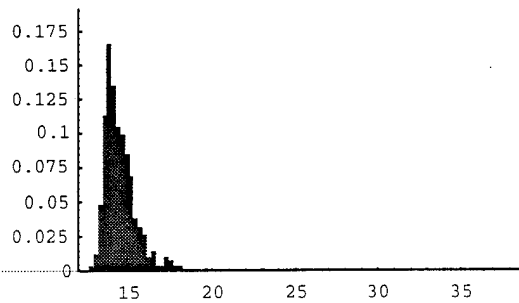
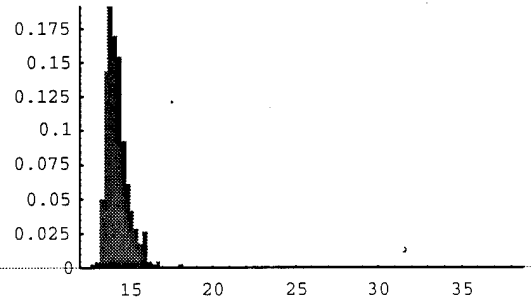
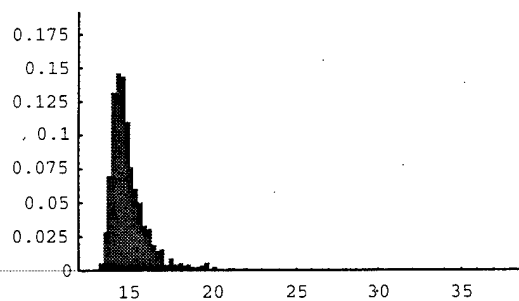
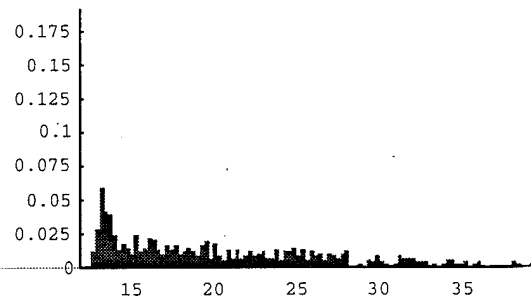
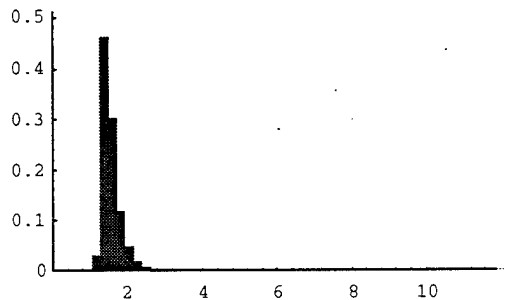
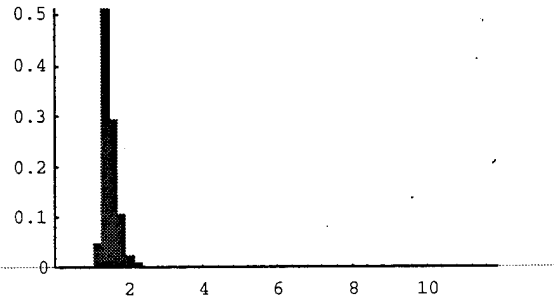
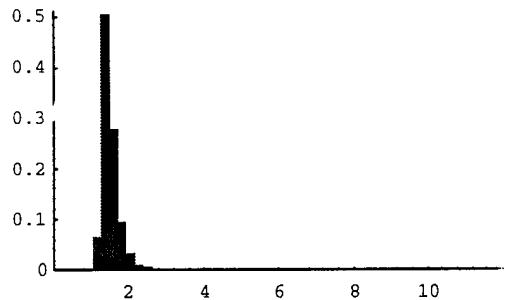
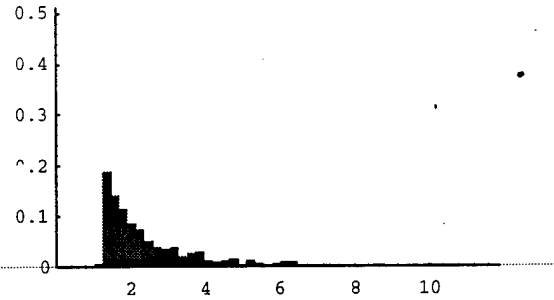




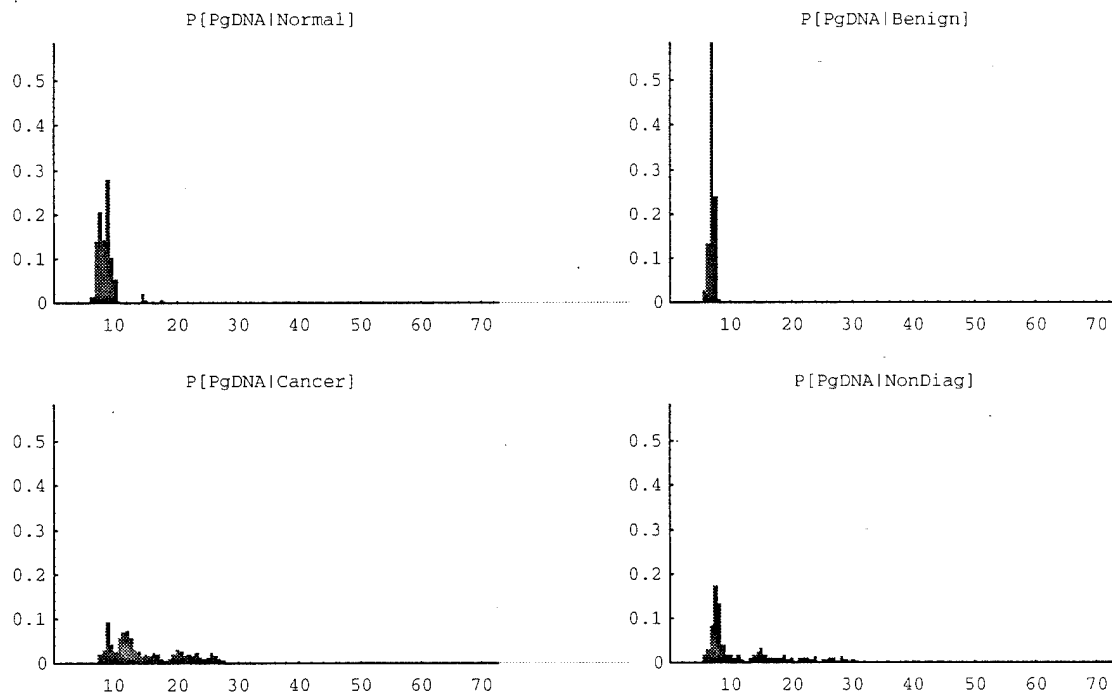
The next feature stands alone with respect to providing a useful measure of the cell size. It is an especially good indicator of NonDiagnostic cells that are due to overlapping cells (because two cells that are touching or slightly overlapping will tend to have a small Minimum Diameter, but a large Maximum Diameter).

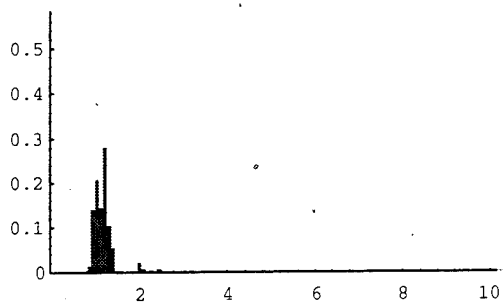
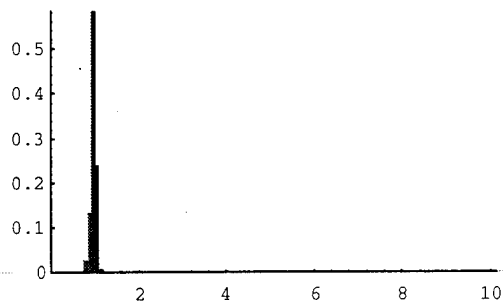
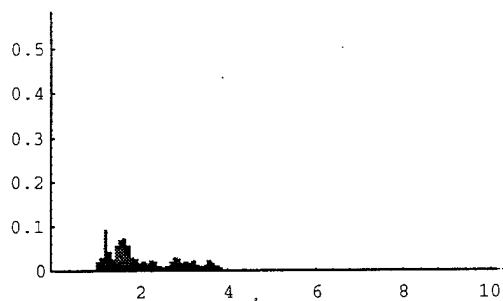
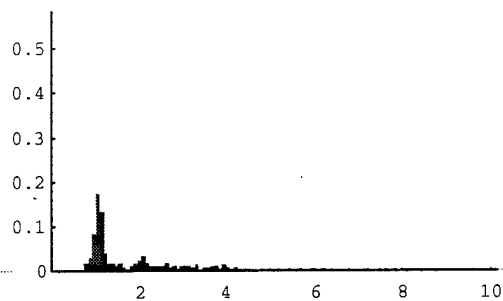
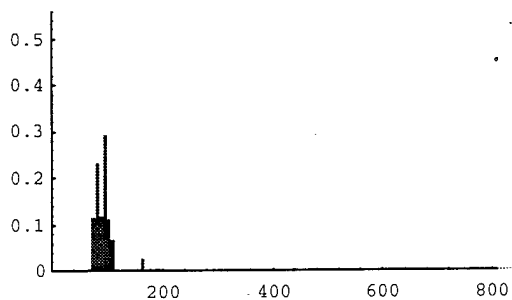
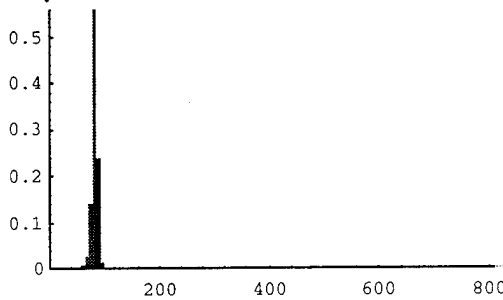


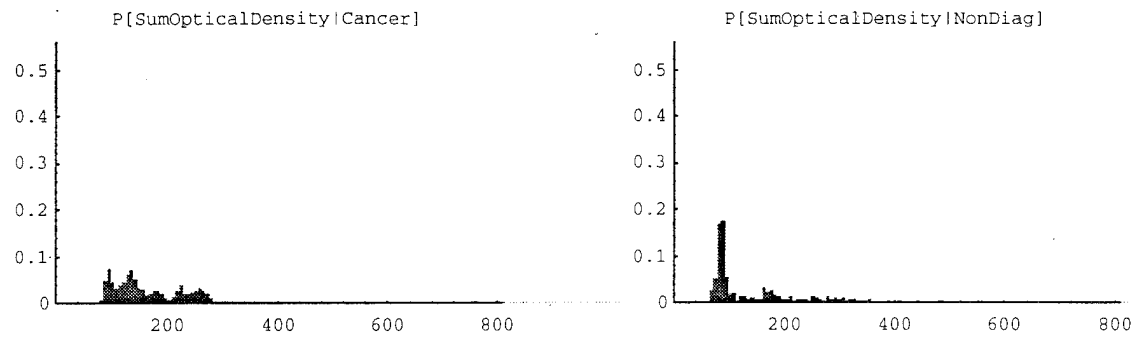
Shape and Elongation are closely related; this shows up in their respective conditional PDFs.

 $P[\text{Shape}|\text{Normal}]$  $P[\text{Shape}|\text{Benign}]$  $P[\text{Shape}|\text{Cancer}]$  $P[\text{Shape}|\text{NonDiag}]$  $P[\text{Elongation}|\text{Normal}]$  $P[\text{Elongation}|\text{Benign}]$  $P[\text{Elongation}|\text{Cancer}]$  $P[\text{Elongation}|\text{NonDiag}]$ 

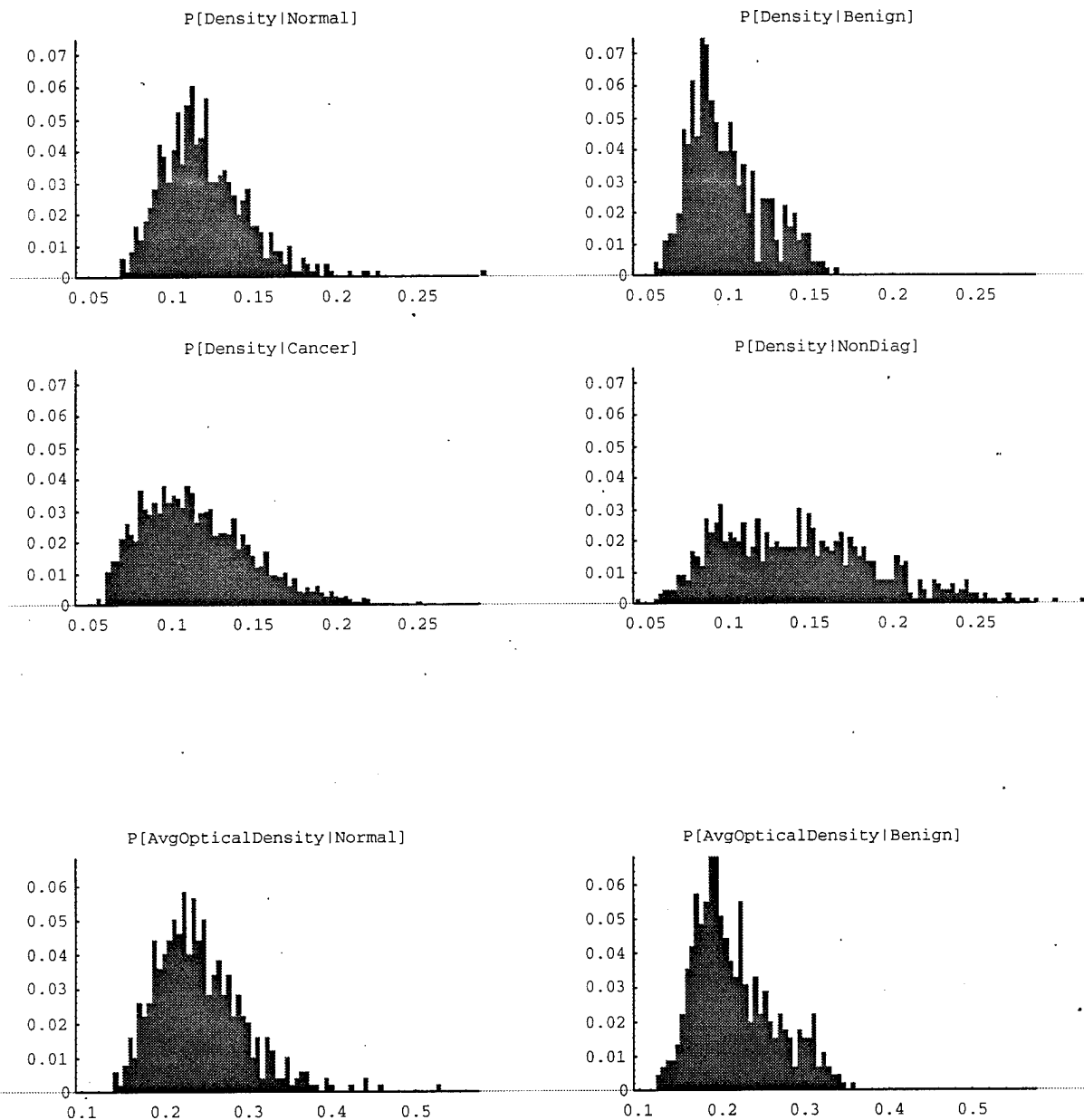
The next group of features measure DNA content. This is due to the way the cell samples are stained in preparation for viewing under the microscope. The stain is sensitive to DNA. DNA content is proportional to the optical density of the cell image. Note that Normal cells and Benign cells are bimodal, with most of the data clustered around the first mode, and a small amount clustered around a mode that is located at twice the value of the first mode. This is due to the fact that such cells have twice as much DNA content just before mitosis (the growth phase at which a cell divides into two separate cells). The feature "DNA Index" makes this especially clear: DNA Index is obtained by normalizing the DNA content by the DNA content of normal cells. Therefore, a normal cell typically has a DNA Index of 1 (typically) or 2 (during cell mitosis). On the other hand, Cancer cells may have a noninteger value of the DNA Index, ranging anywhere from 1 up to and beyond 2. Of course, the same holds true of the Nondiagnostic class, which contains objects which correspond to several cells, which themselves may or may not be cancerous (therefore, while the Nondiagnostic PDF exhibits strong peaks at integer multiples of the first peak, there are also many cells that lie between the peaks and beyond). Note that DNA content is calculated directly from the Summed Optical Density.

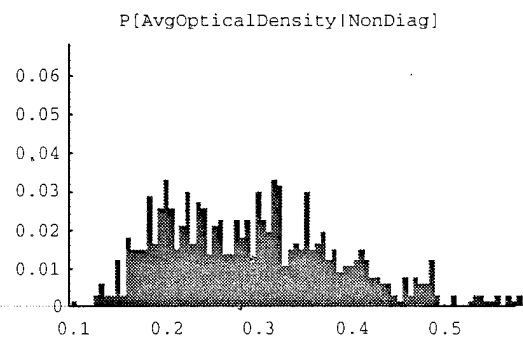
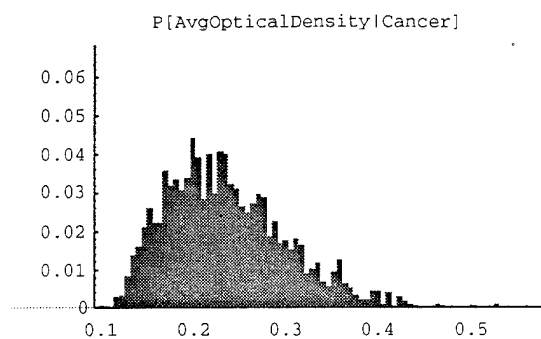


$P[\text{DNAIndex}|\text{Normal}]$  $P[\text{DNAIndex}|\text{Benign}]$  $P[\text{DNAIndex}|\text{Cancer}]$  $P[\text{DNAIndex}|\text{NonDiag}]$  $P[\text{SumOpticalDensity}|\text{Normal}]$  $P[\text{SumOpticalDensity}|\text{Benign}]$ 

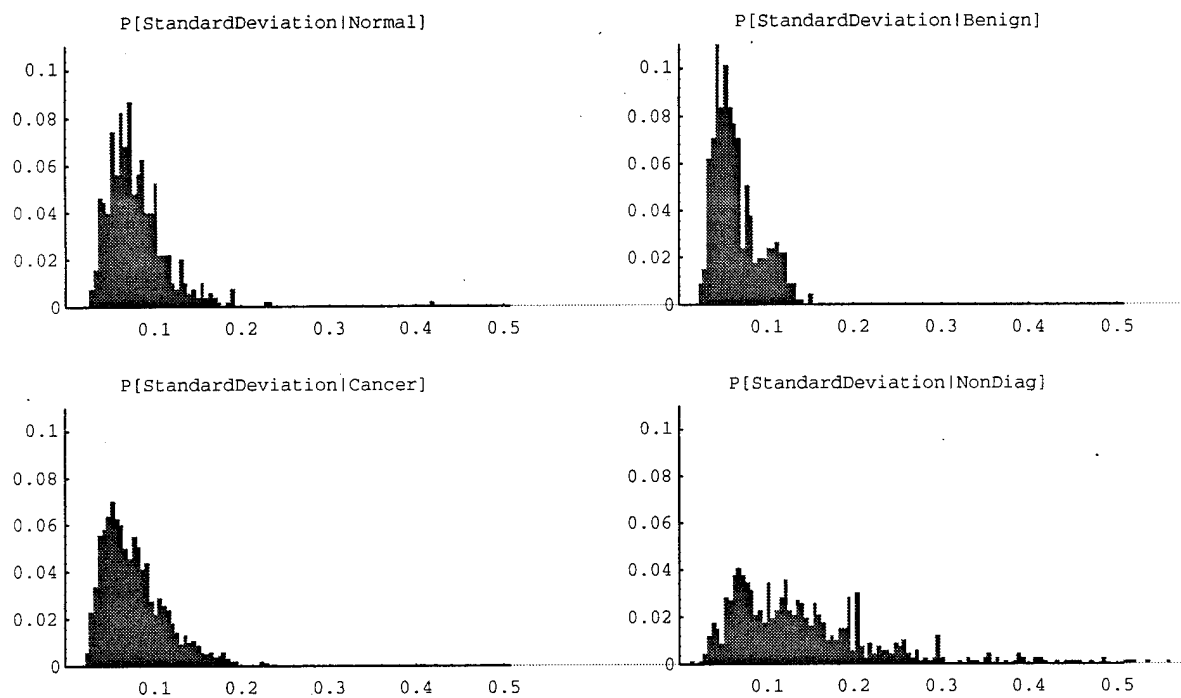


Density and Average Optical Density are considered measures of morphometry, but actually measure aspects of the cell image texture. Density here means the total DNA content divided by cell area. Average Optical Density measures essentially the same thing.

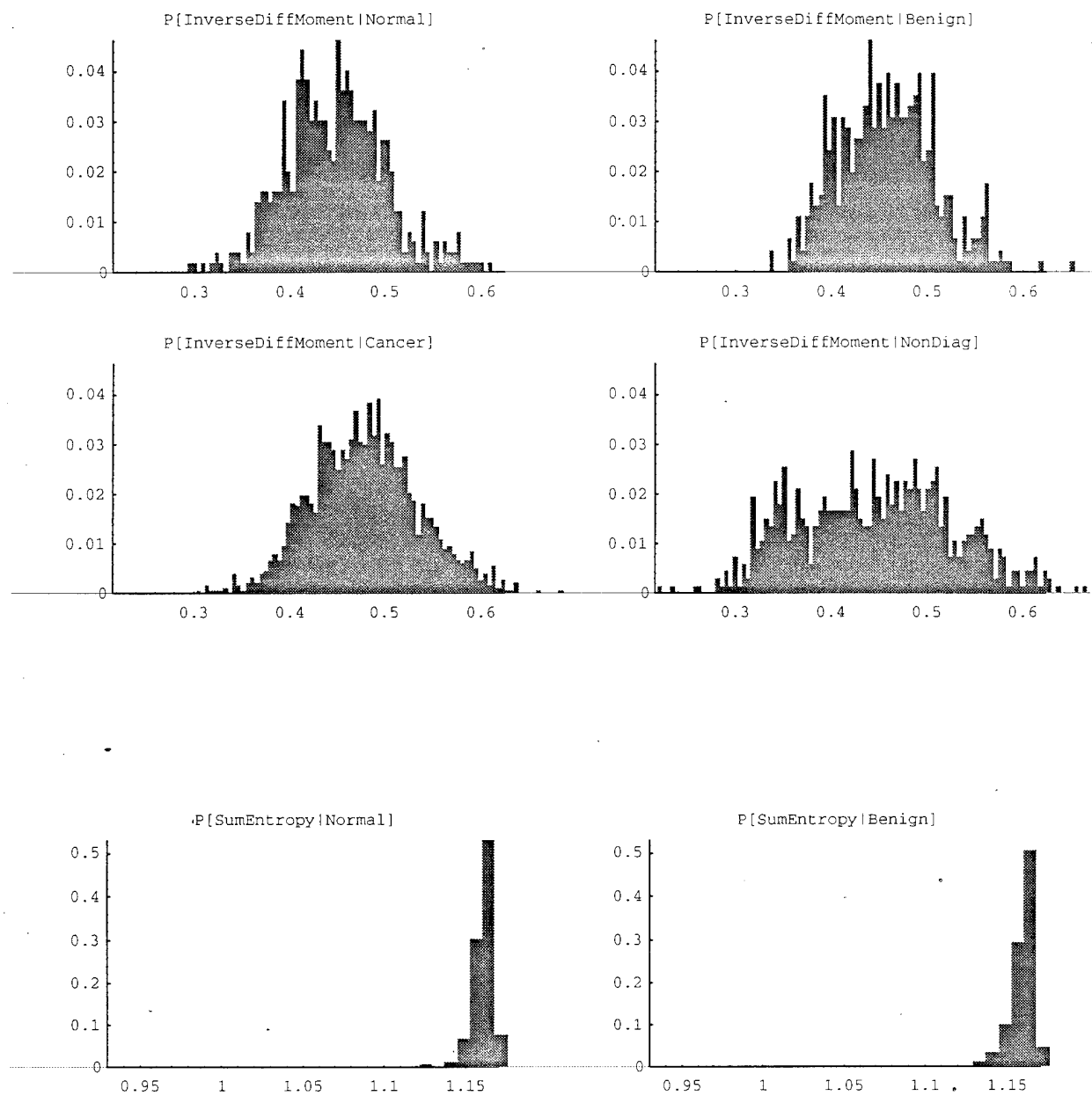


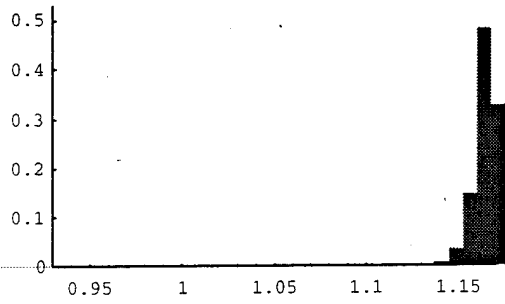
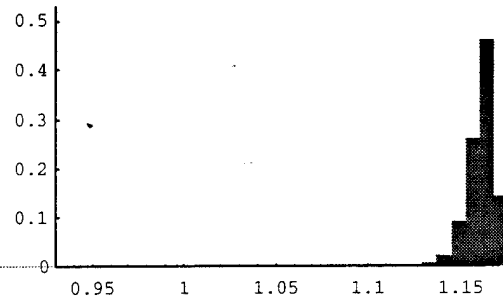


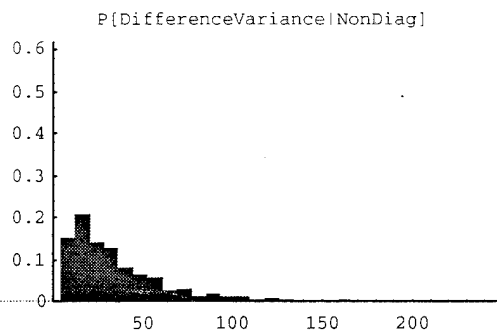
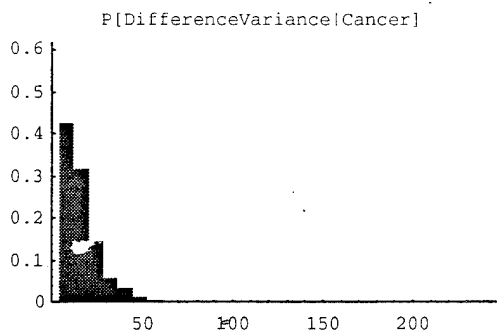
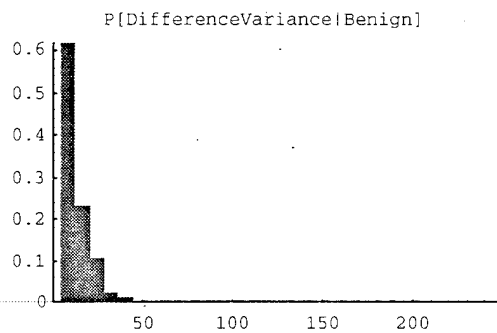
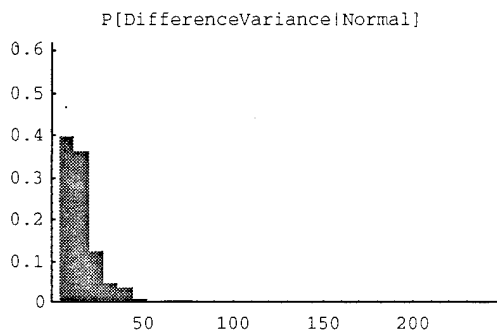
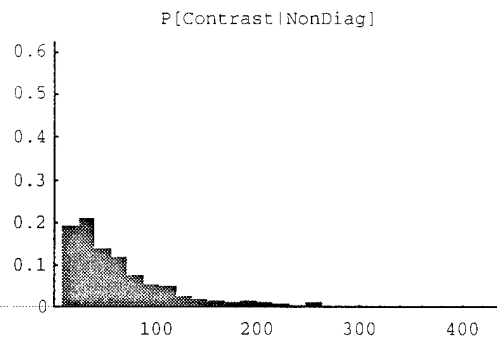
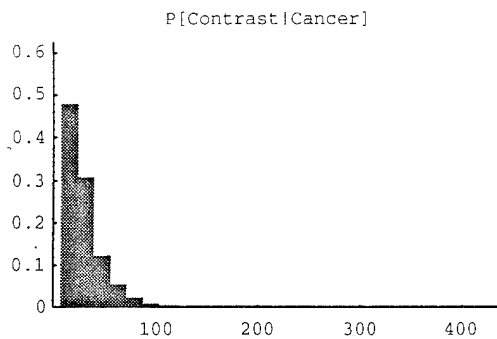
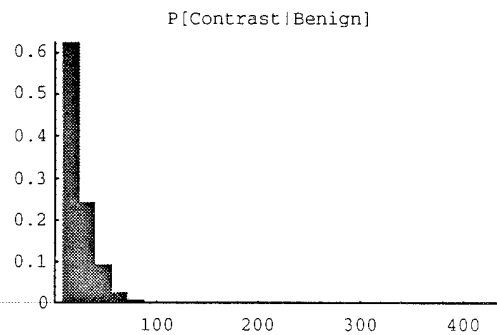
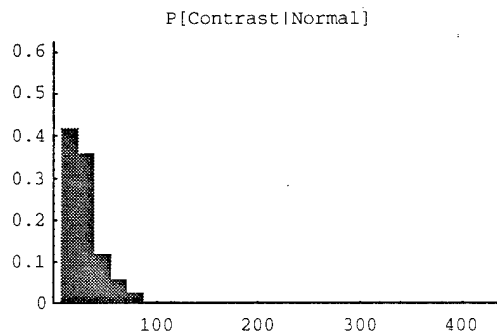
Standard Deviation is a hybrid between the morphometric features (displayed above) and the texture features (to follow below).

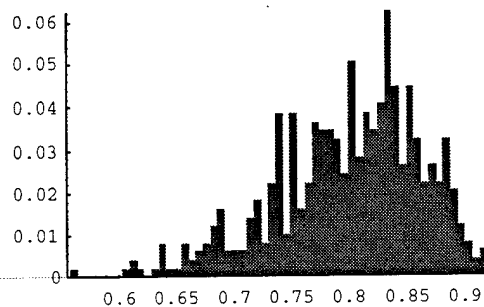
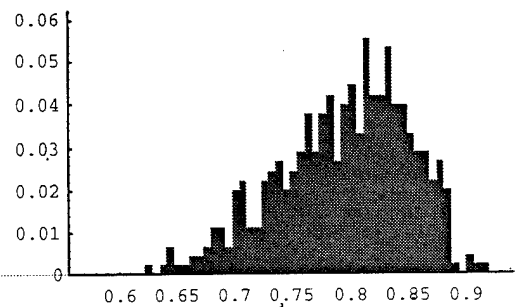
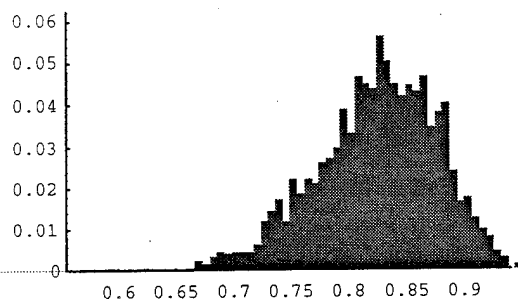
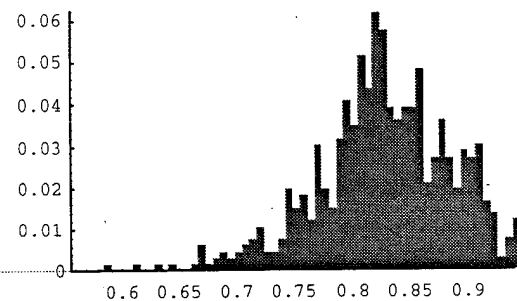
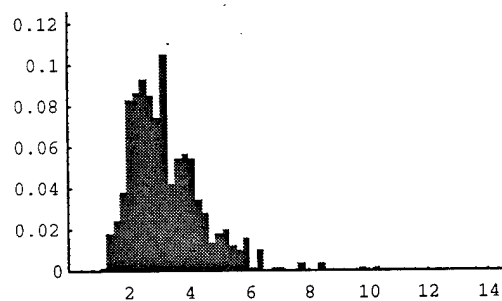
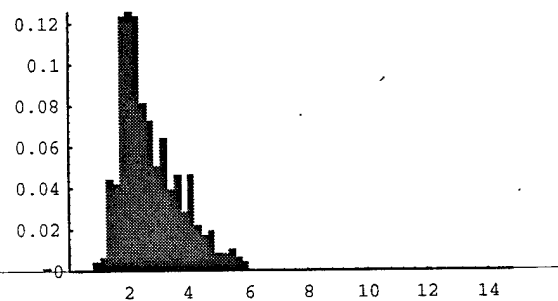


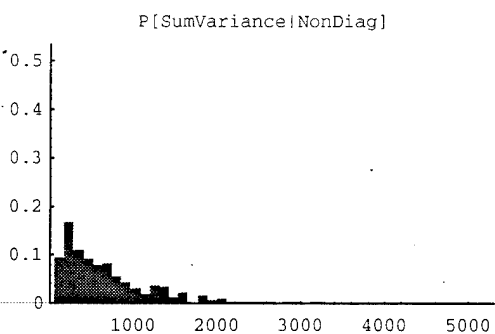
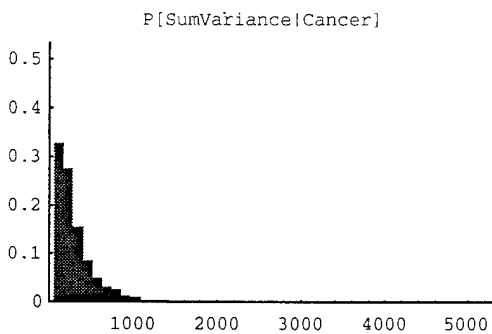
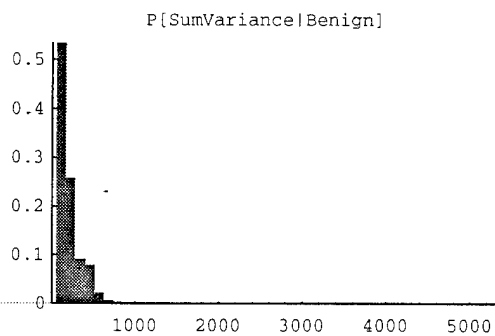
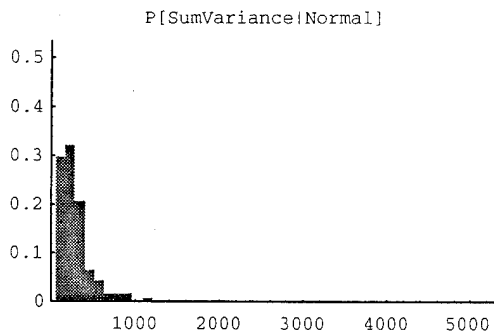
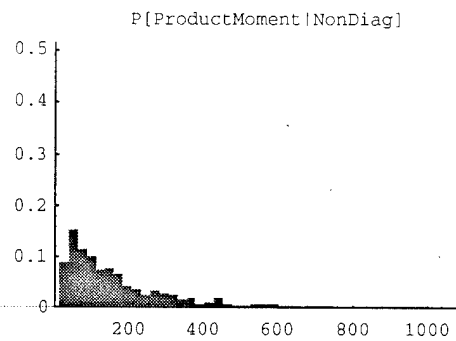
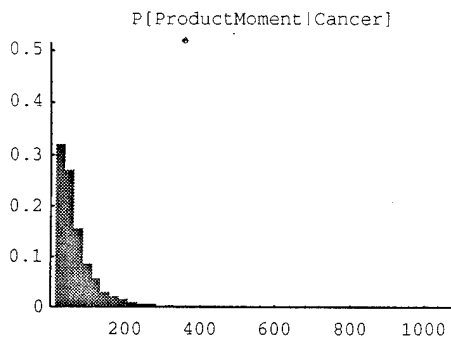
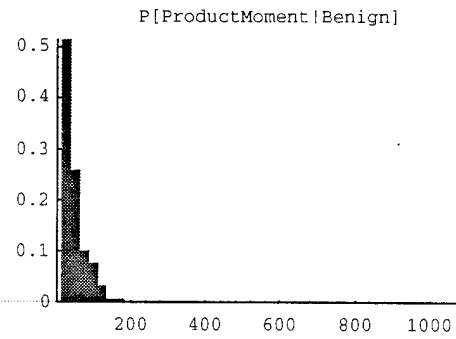
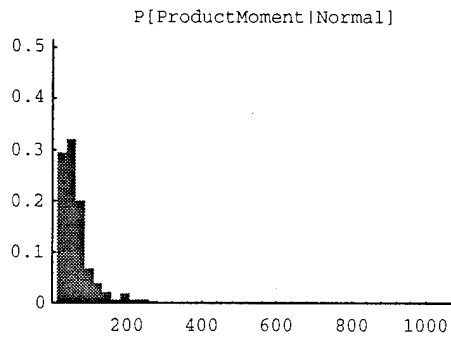
The remaining PDFs are texture features. Like the morphometric features displayed above, they are also grouped according to visual similarity.

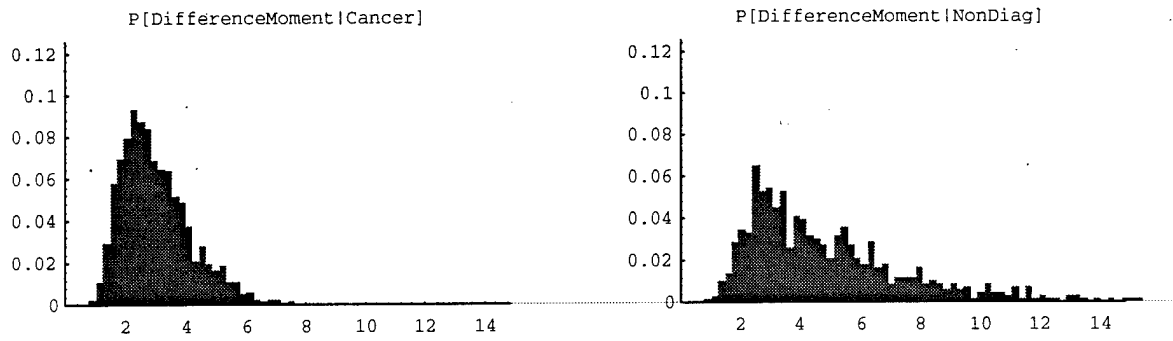


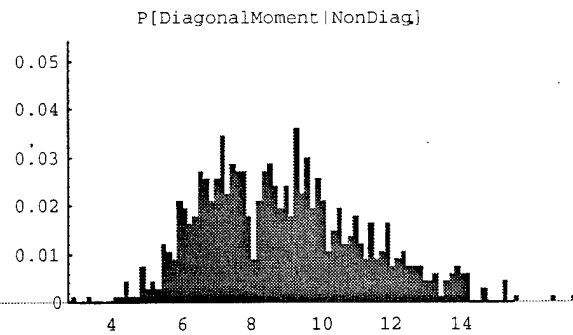
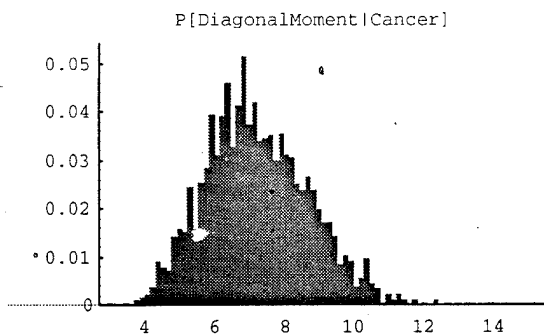
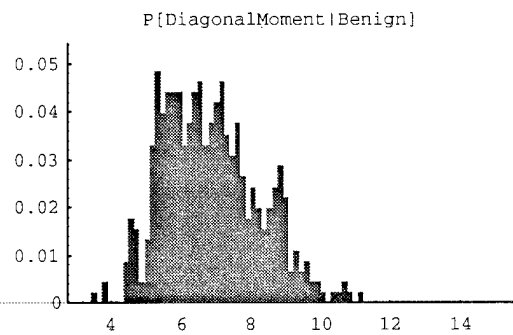
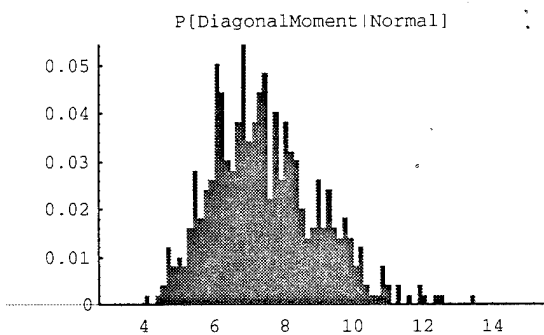
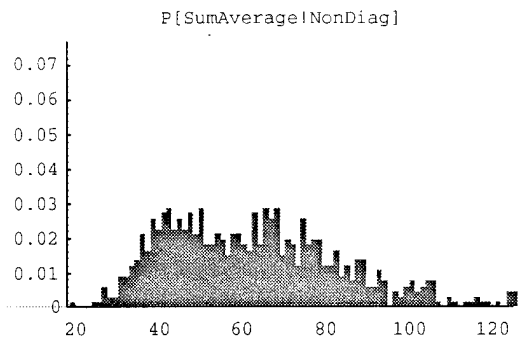
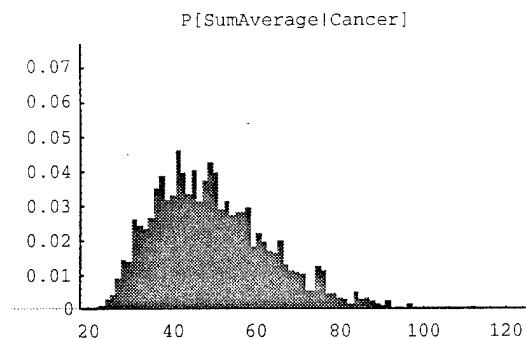
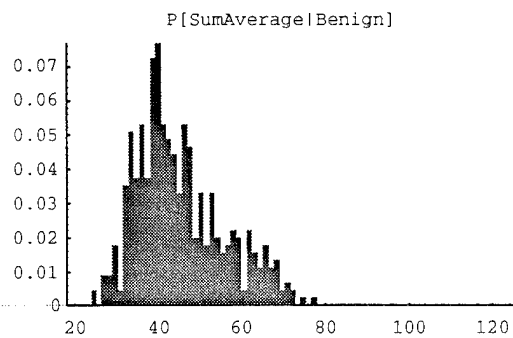
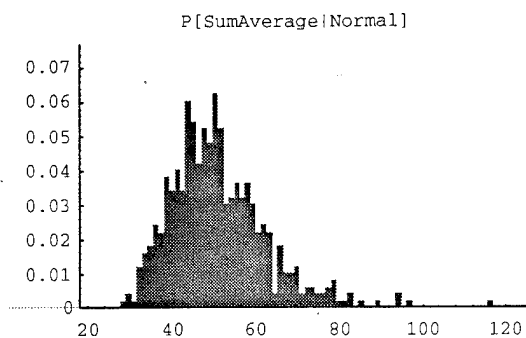
$P[\text{SumEntropy}|\text{Cancer}]$  $P[\text{SumEntropy}|\text{NonDiag}]$ 



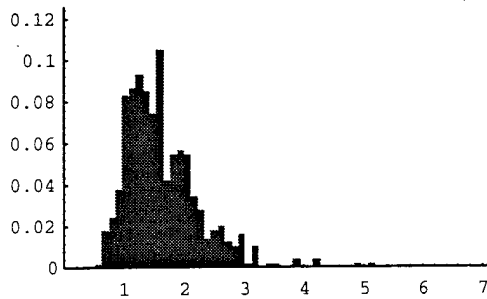
$P[\text{Correlation}|\text{Normal}]$  $P[\text{Correlation}|\text{Benign}]$  $P[\text{Correlation}|\text{Cancer}]$  $P[\text{Correlation}|\text{NonDiag}]$  $P[\text{DifferenceMoment}|\text{Normal}]$  $P[\text{DifferenceMoment}|\text{Benign}]$ 



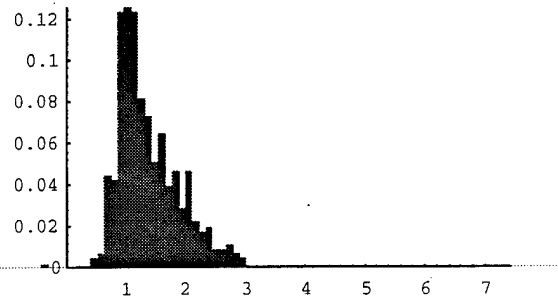




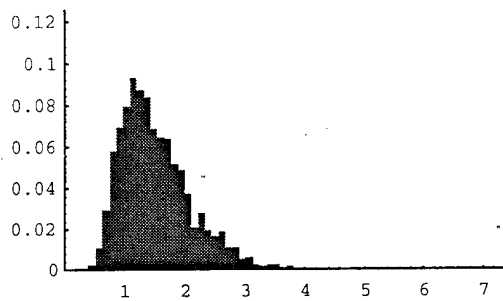
P[SecondDiagonalMoment|Normal]



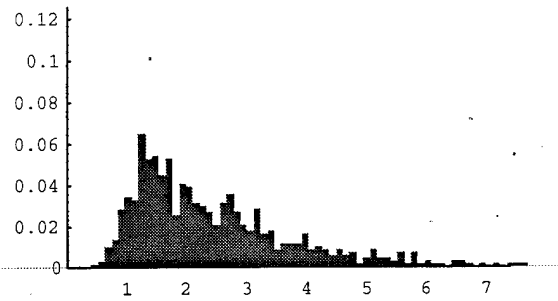
P[SecondDiagonalMoment|Benign]



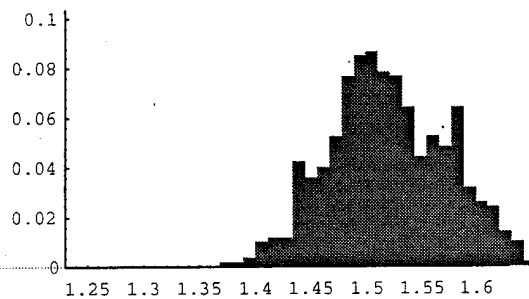
P[SecondDiagonalMoment|Cancer]



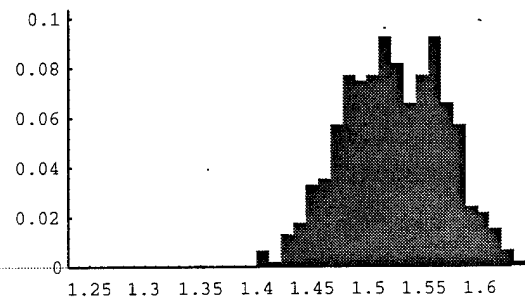
P[SecondDiagonalMoment|NonDiag]



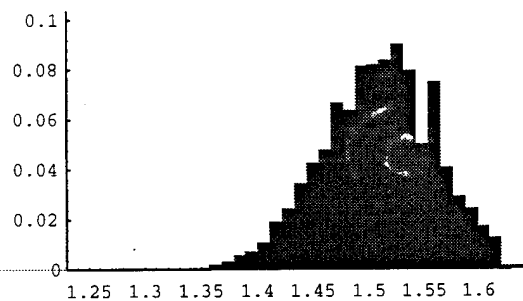
P[Entropy|Normal]



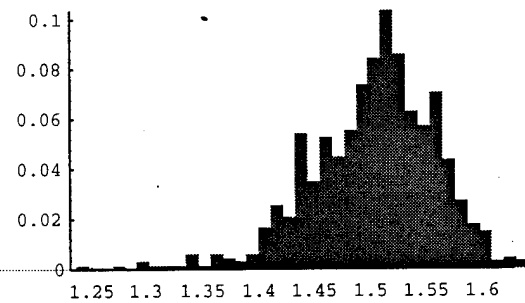
P[Entropy|Benign]

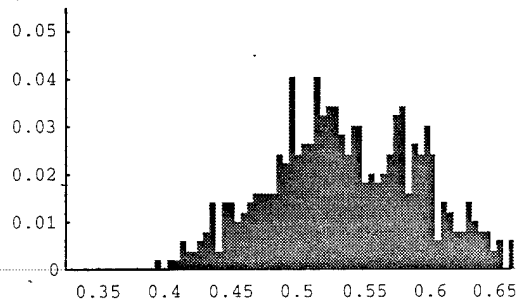
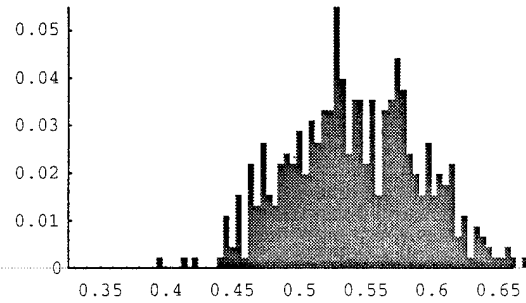
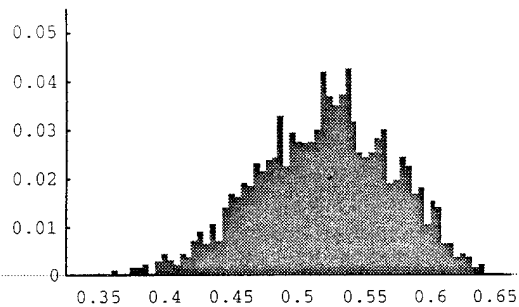
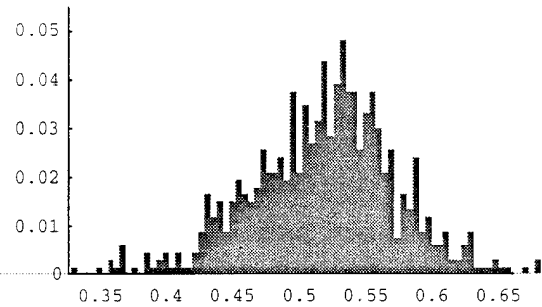
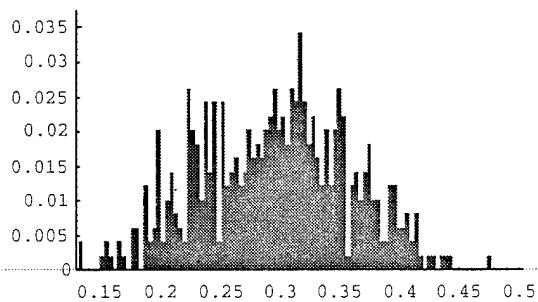
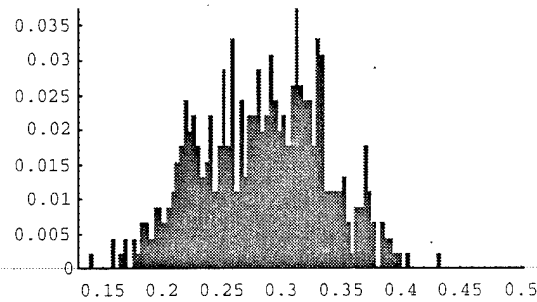
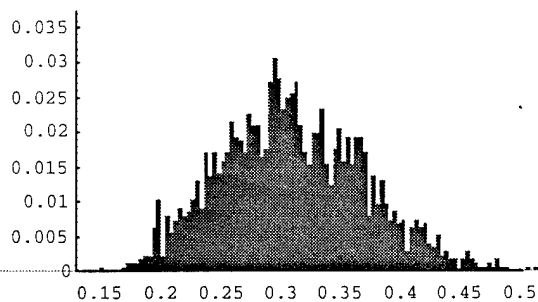
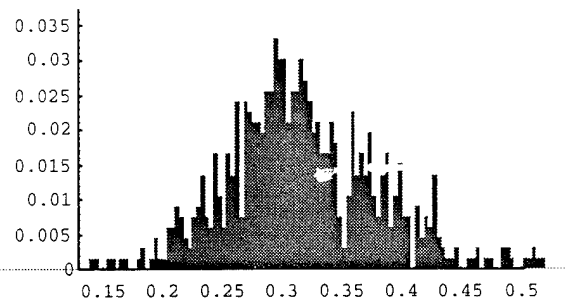


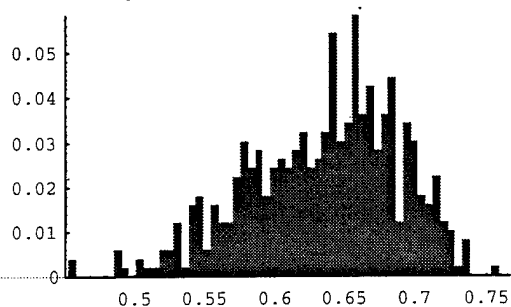
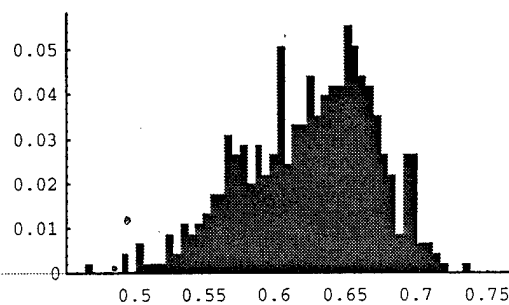
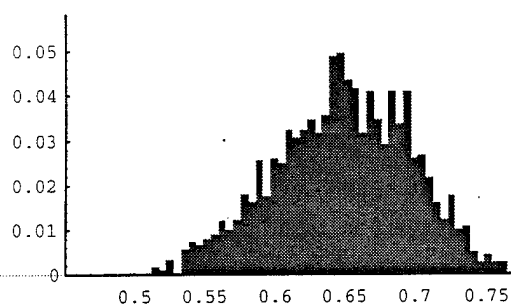
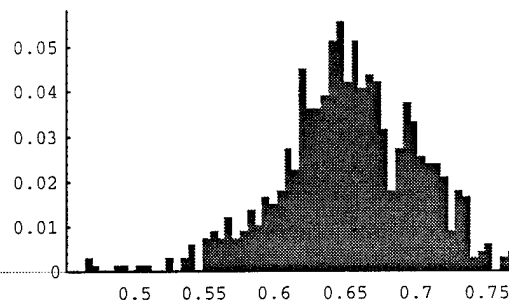
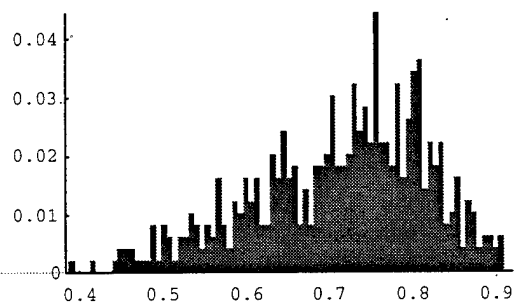
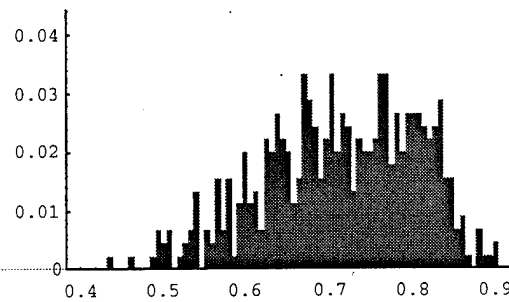
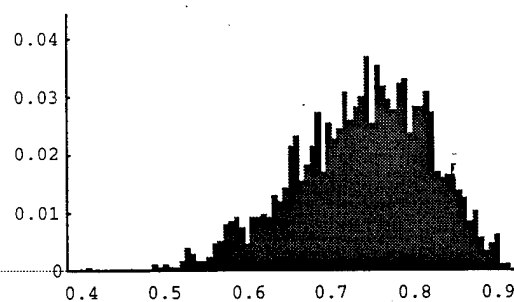
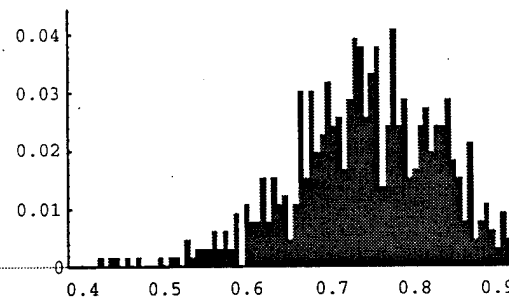
P[Entropy|Cancer]



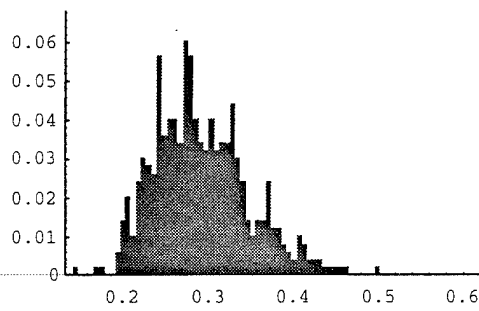
P[Entropy|NonDiag]



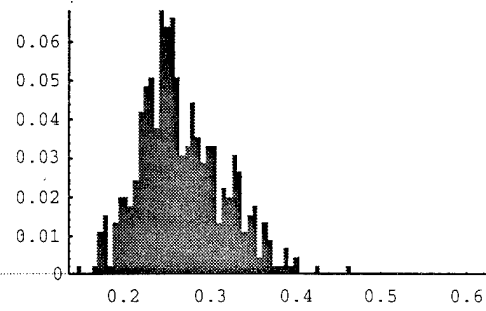
$P[\text{DifferenceEntropy}|\text{Normal}]$  $P[\text{DifferenceEntropy}|\text{Benign}]$  $P[\text{DifferenceEntropy}|\text{Cancer}]$  $P[\text{DifferenceEntropy}|\text{NonDiag}]$  $P[\text{InformationMeasureA}|\text{Normal}]$  $P[\text{InformationMeasureA}|\text{Benign}]$  $P[\text{InformationMeasureA}|\text{Cancer}]$  $P[\text{InformationMeasureA}|\text{NonDiag}]$ 

$P[\text{InformationMeasureB}|\text{Normal}]$  $P[\text{InformationMeasureB}|\text{Benign}]$  $P[\text{InformationMeasureB}|\text{Cancer}]$  $P[\text{InformationMeasureB}|\text{NonDiag}]$  $P[\text{MaximalCorrelationCoeff}|\text{Normal}]$  $P[\text{MaximalCorrelationCoeff}|\text{Benign}]$  $P[\text{MaximalCorrelationCoeff}|\text{Cancer}]$  $P[\text{MaximalCorrelationCoeff}|\text{NonDiag}]$ 

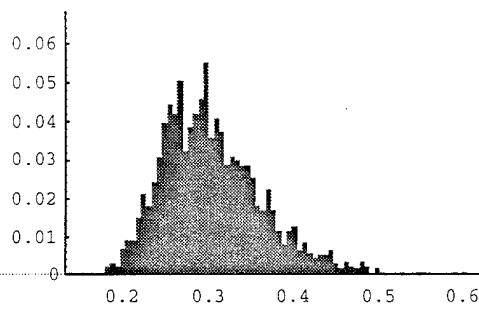
P[CoeffOfVariation|Normal]



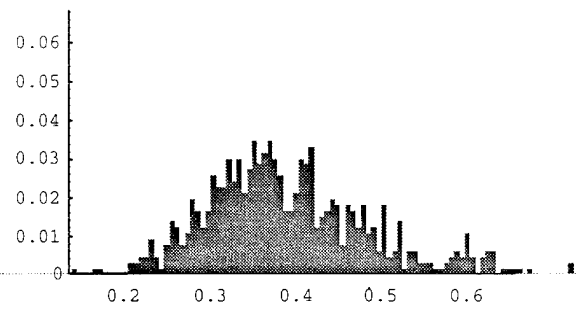
P[CoeffOfVariation|Benign]



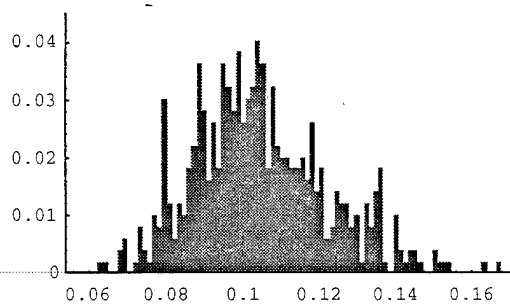
P[CoeffOfVariation|Cancer]



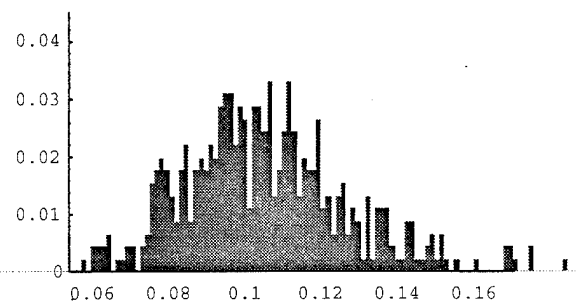
P[CoeffOfVariation|NonDiag]



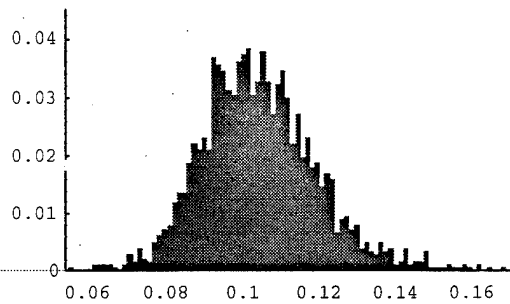
P[PeakTransitionProb|Normal]



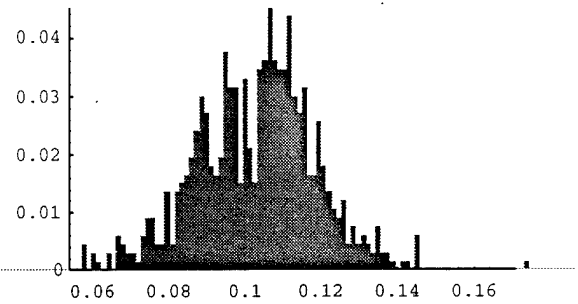
P[PeakTransitionProb|Benign]



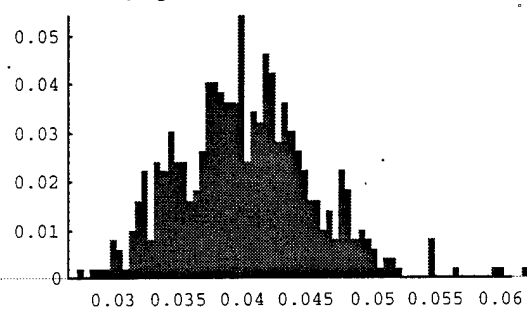
P[PeakTransitionProb|Cancer]



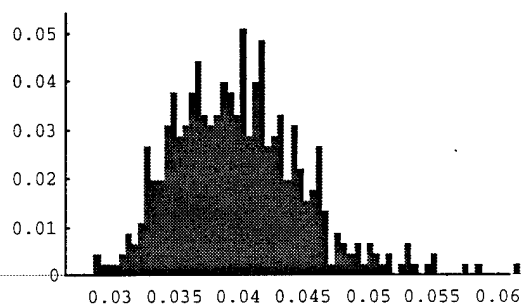
P[PeakTransitionProb|NonDiag]



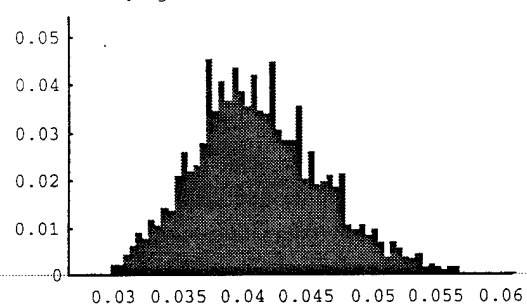
P[AngularSecondMoment|Normal]



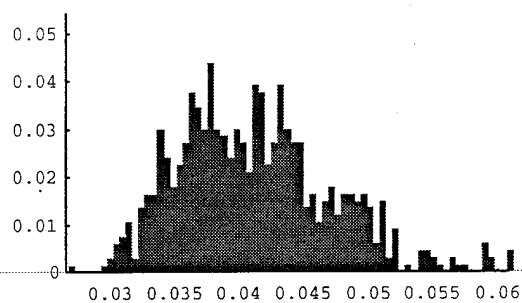
P[AngularSecondMoment|Benign]



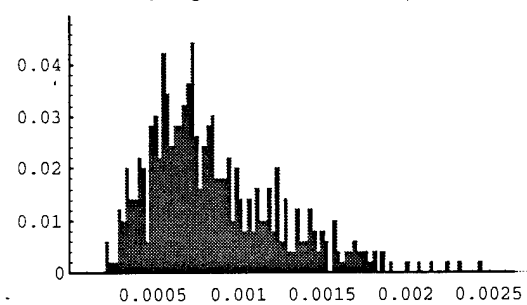
P[AngularSecondMoment|Cancer]



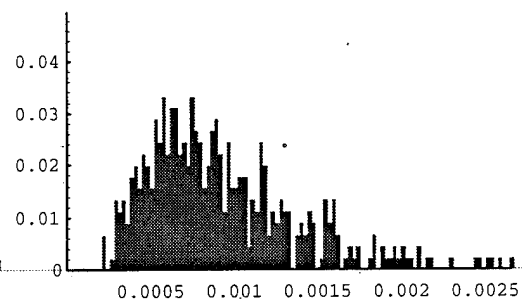
P[AngularSecondMoment|NonDiag]



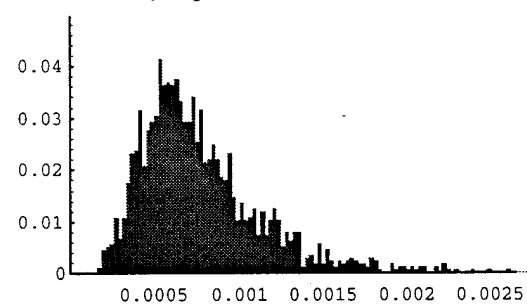
P[DiagonalVariance|Normal]



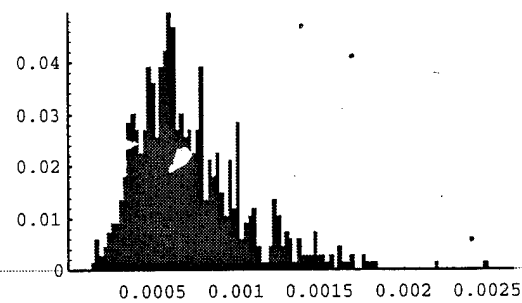
P[DiagonalVariance|Benign]

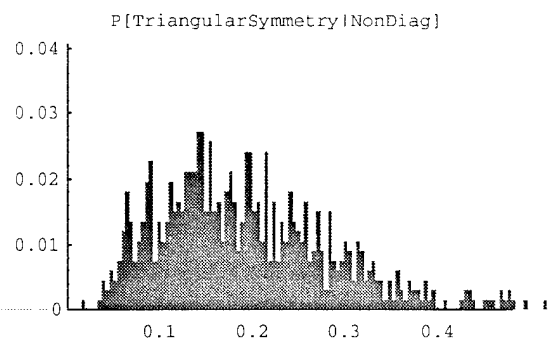
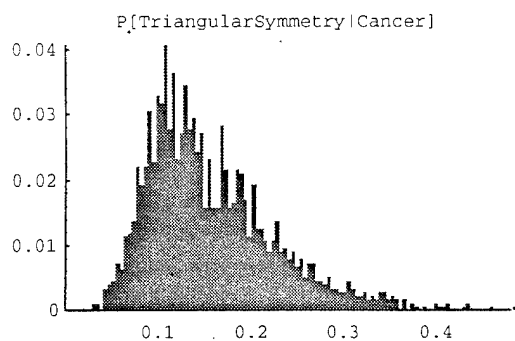
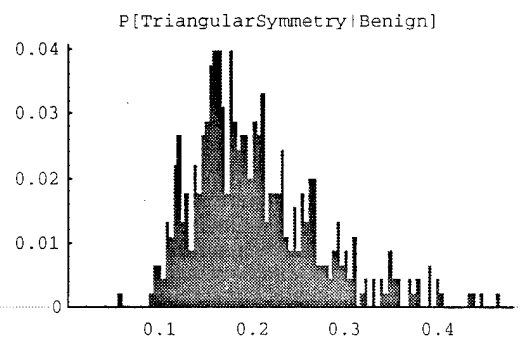
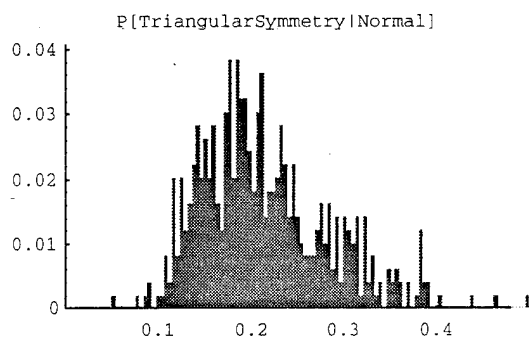


P[DiagonalVariance|Cancer]



P[DiagonalVariance|NonDiag]





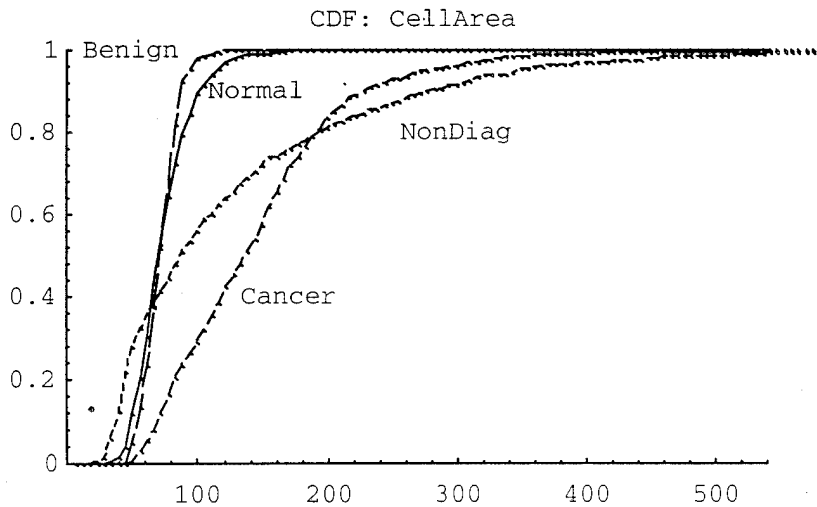
THE CONDITIONAL CDFS

Once we became familiar with viewing conditional PDFs, we then generated for each feature the conditional cumulative distribution function (CDFs) given each class. This was useful for two reasons:

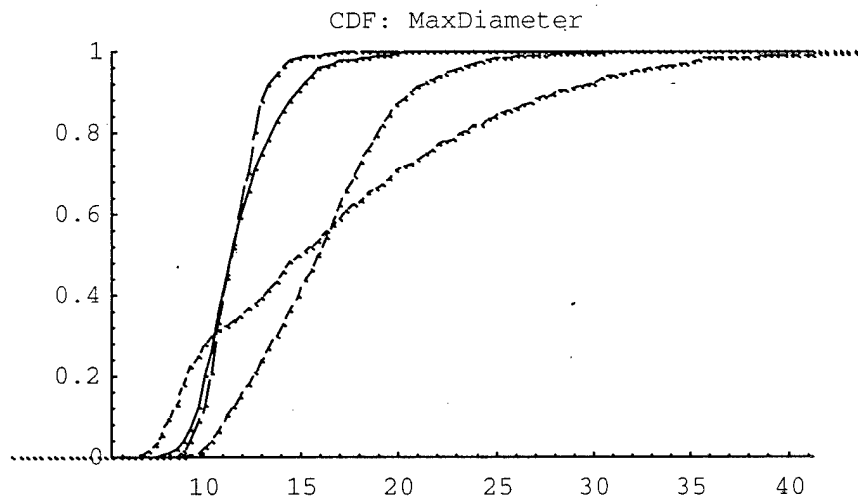
1. CDFs are less sensitive to choice of bin size necessary to construct the histogram estimate of a PDF.
2. Once one is familiar with viewing CDFs, it is easy to quickly spot differences and similarities among a set of CDFs viewed simultaneously (whereas differences or similarities may not be as apparent among a set of histograms).

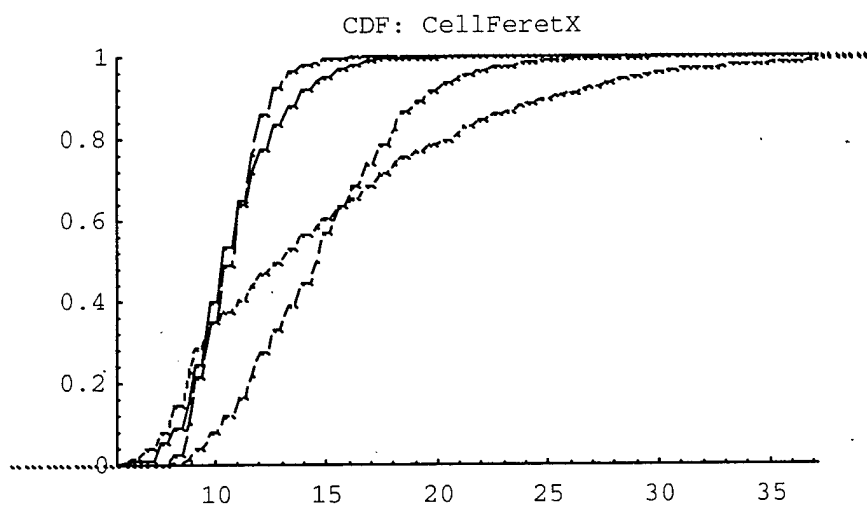
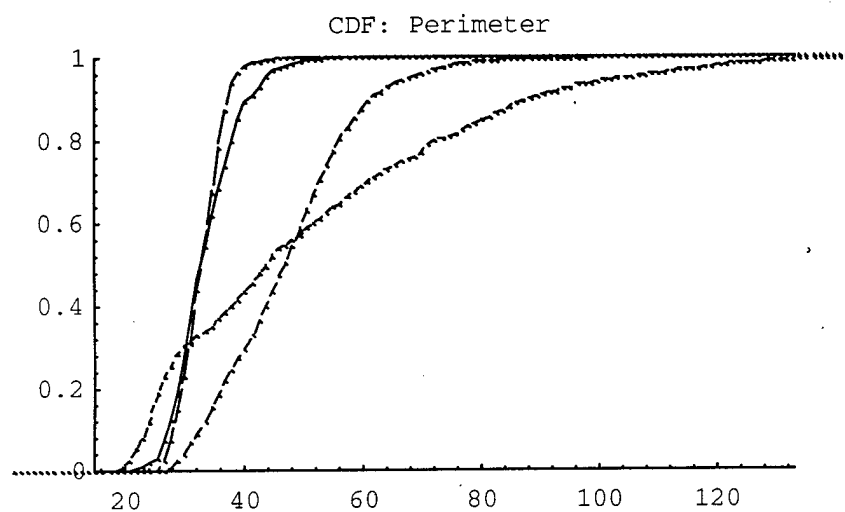
The same rules were used to calculate the range of the horizontal axis. Therefore the reader may compare the CDFs with the PDFs given above. For example, the reader may discern the presence of the second peak present in the DNA Index PDF for the Normal class by the notch in the corresponding conditional CDF curve.

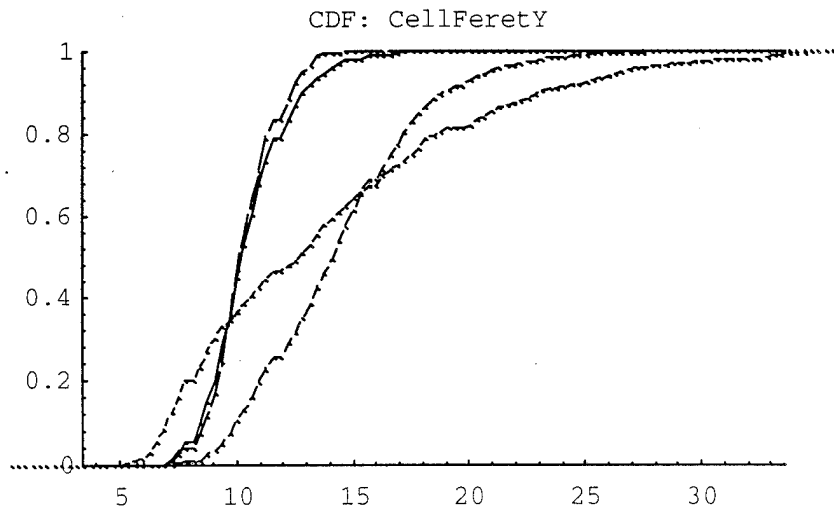
The first graph contains labels for the lines. Subsequent plots use the same types of lines: solid for Normal, broken for Benign, dashed for Cancer, and dotted for NonDiagnostic.



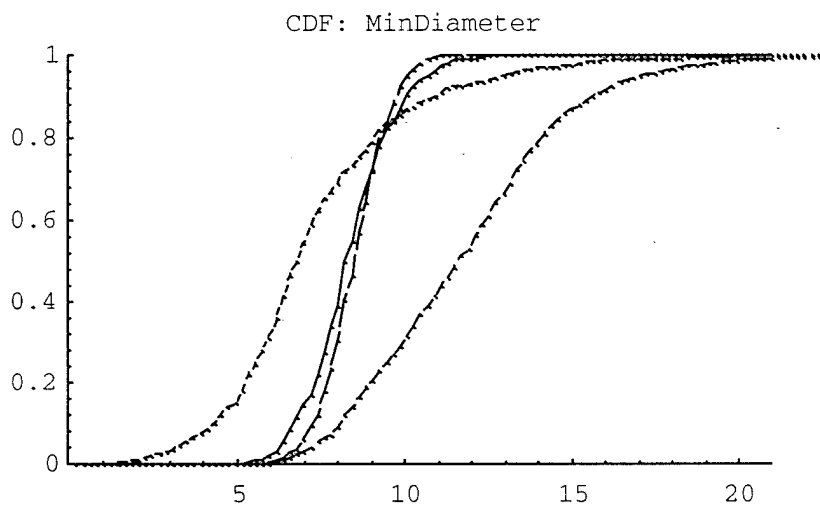
The following CDF is of Maximum Diameter. The Cancer and Nondiagnostic classes are spread towards larger values. The Nondiagnostic has a larger proportion of larger cells (recall that Nondiagnosics can be comprised of several Cancer cells, therefore Nondiagnosics will always be larger than Cancer cells on average) as well as a larger proportion of smaller cells (note that the Nondiagnostic curve rises up quickly sooner than any of the other curves). This is because the Nondiagnostic class also contains Lymphocytes and Neutrophils, which are typically much smaller than other types of cells.



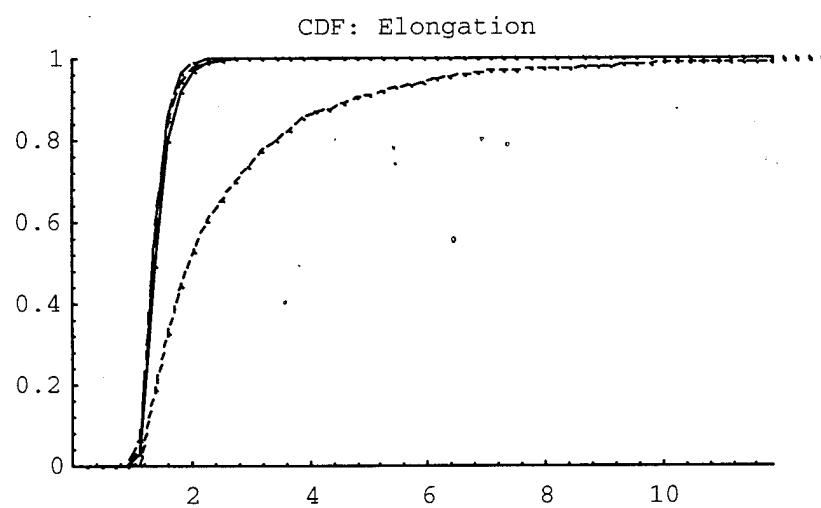
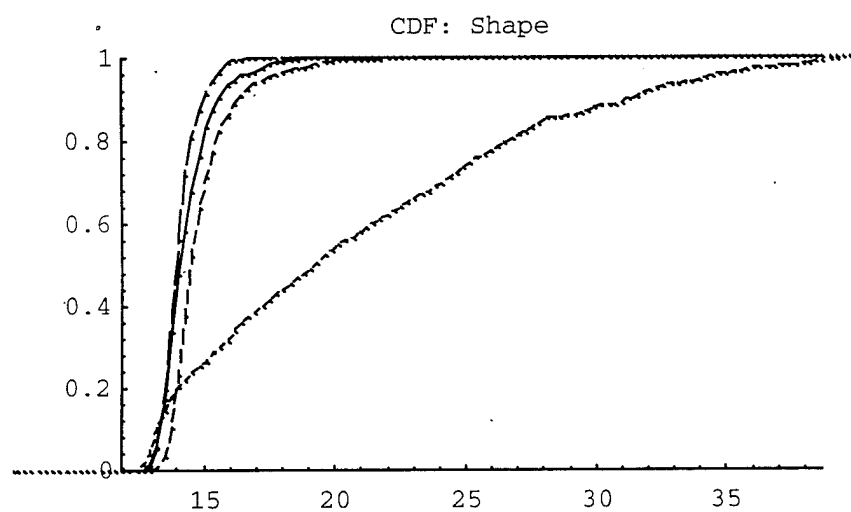




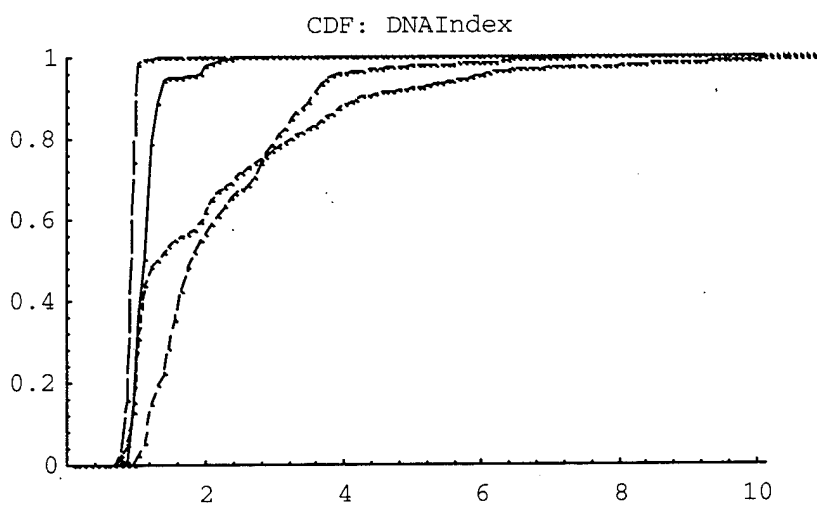
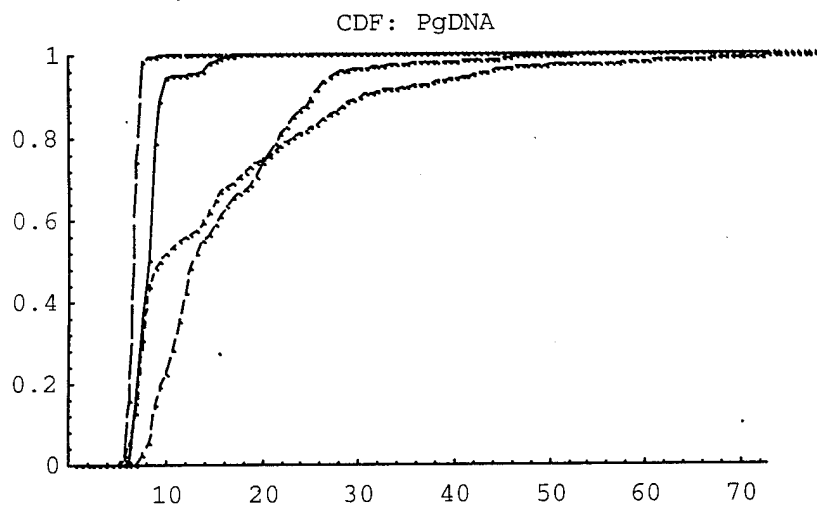
Nondiagnostic cells can have an extremely small Minimum Diameter, due either to the fact the Nondiagnostic object corresponds to a small cell, or to the fact that it can be comprised of a clump of two or more cells which barely touch at some point.

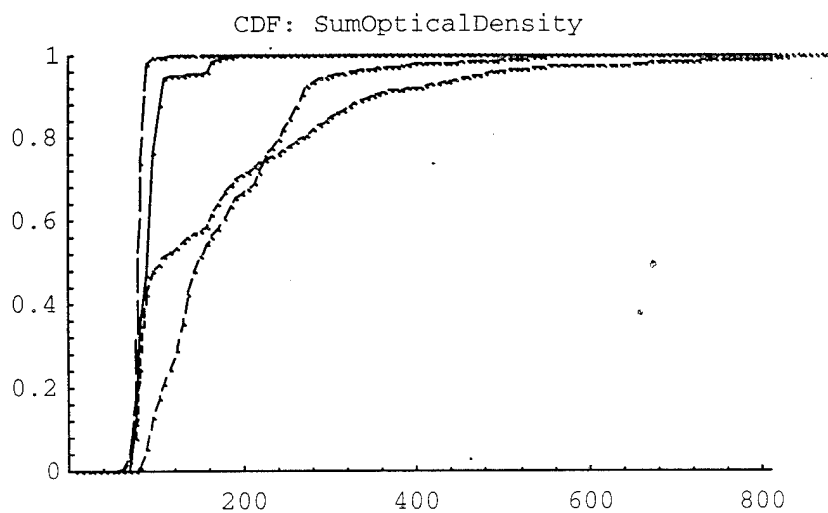


Shape and Elongation are very similar (Shape equals squared Perimeter divided by Area, Elongation equals Max Diameter divided by Minimum Diameter).

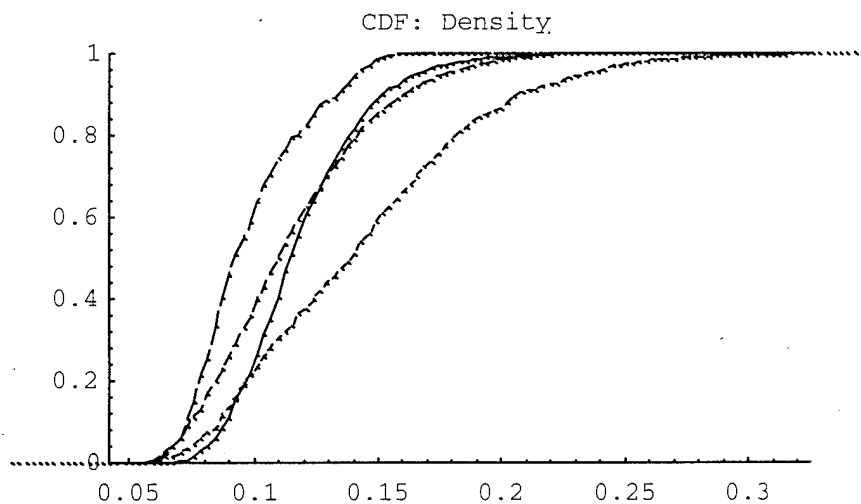


Note the little notch in the Normal curve for each of the following 3 CDFs: this corresponds to a small number of cells that happen to be about to divide in two (undergoing cell mitosis) and therefore contain twice the normal DNA content. Nondiagnostic and Cancer both have significantly higher DNA content overall.

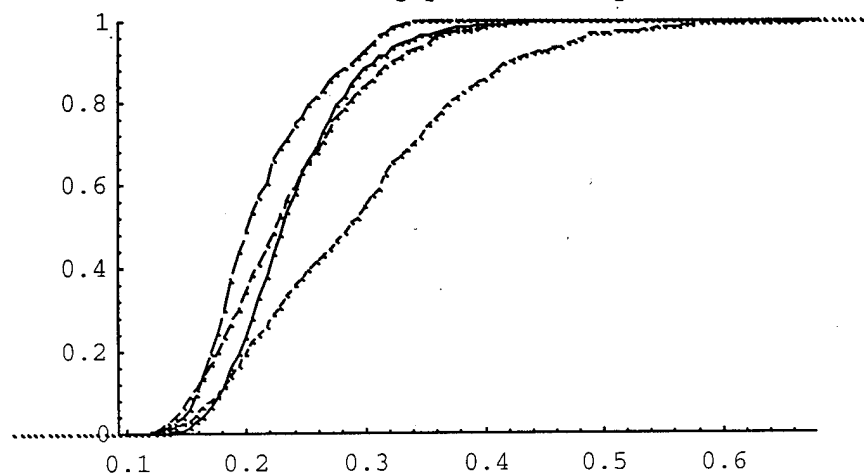




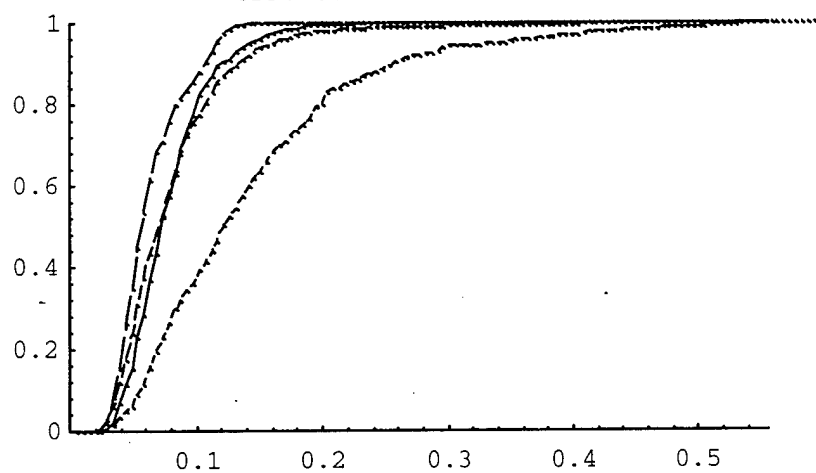
Here is a rare case of one feature standing out with respect to Benign cells (the curve that rises most quickly). It also stands out with respect to Nondiagnostics (the curve that rises most slowly).



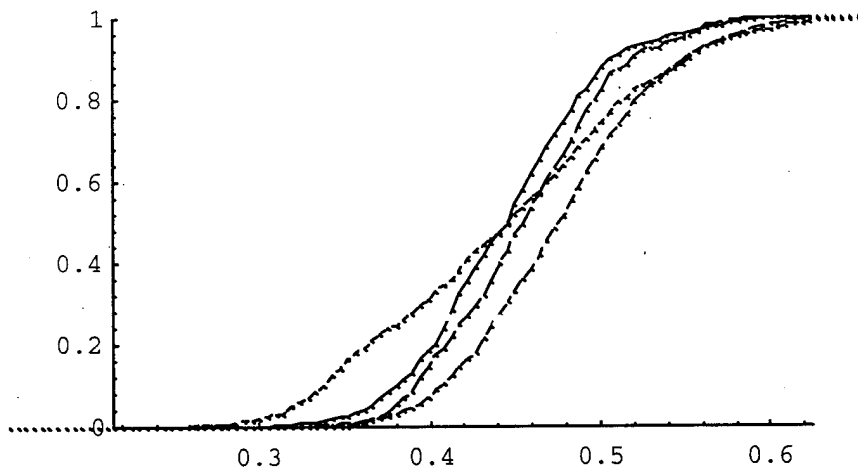
CDF: AvgOpticalDensity



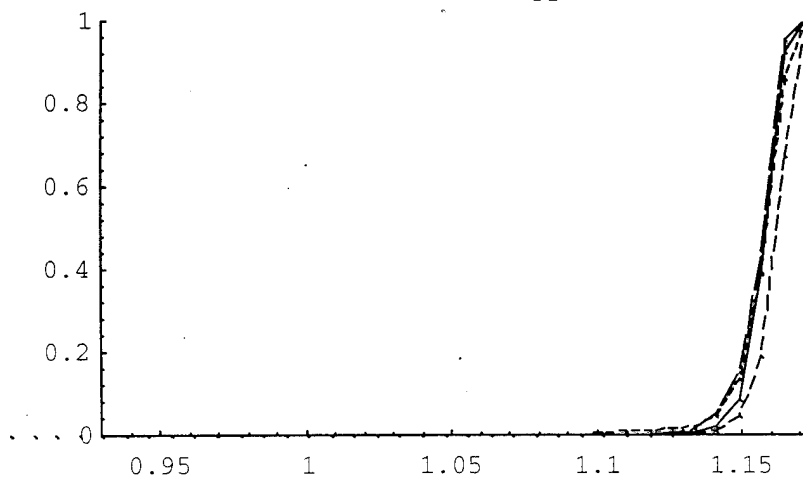
CDF: StandardDeviation



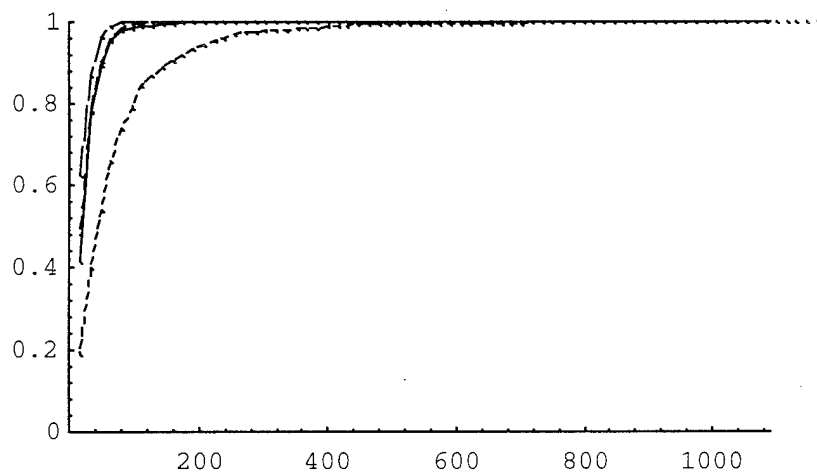
CDF: InverseDiffMoment



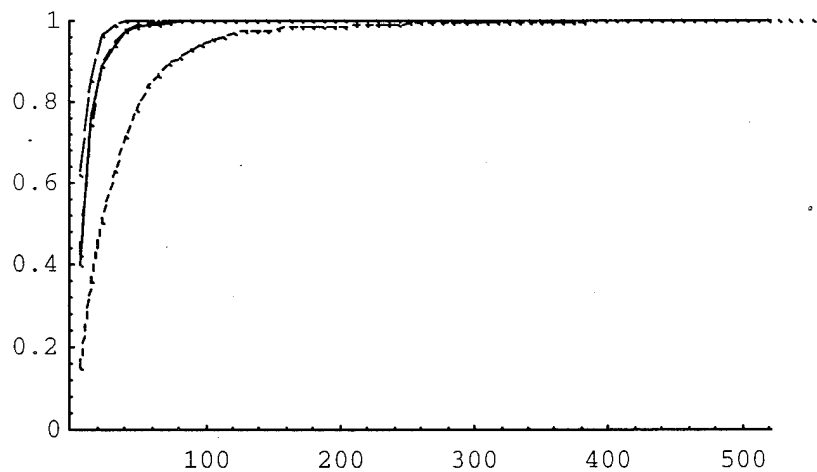
CDF: SumEntropy



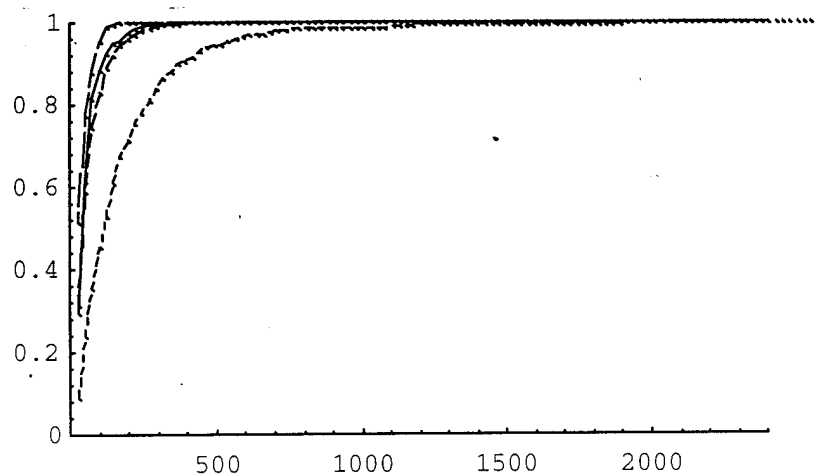
CDF: Contrast



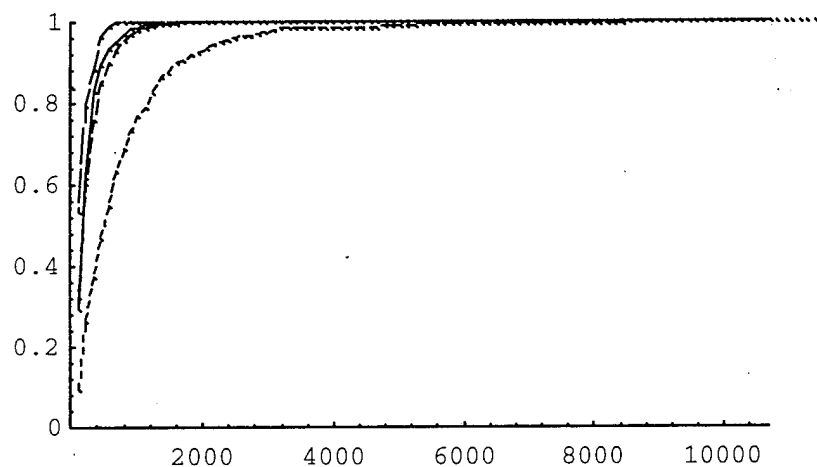
CDF: DifferenceVariance



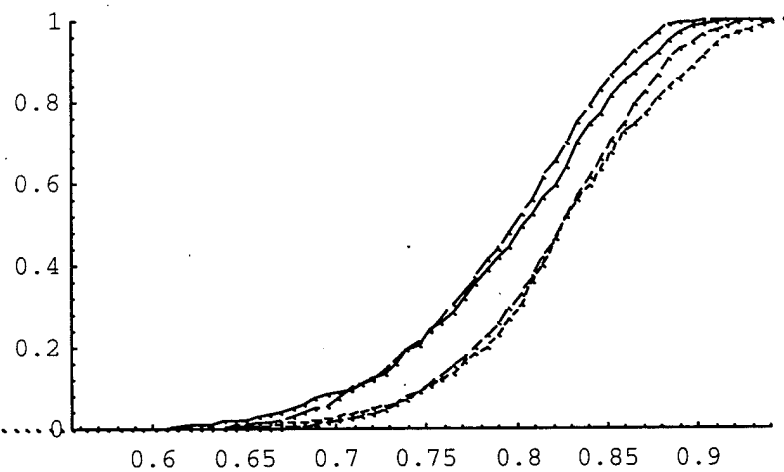
CDF: ProductMoment

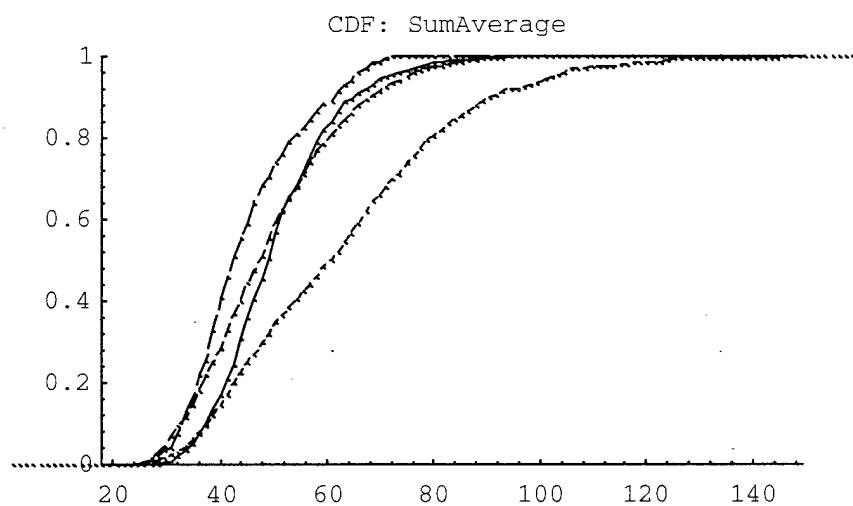
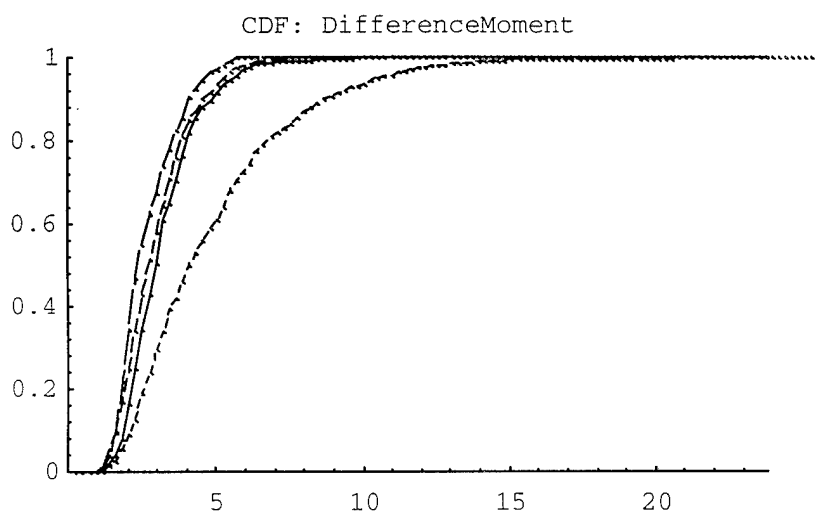


CDF: SumVariance

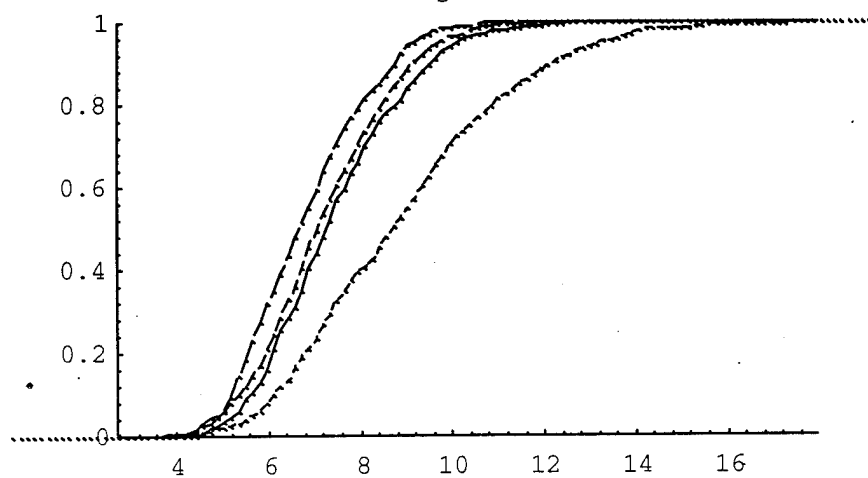


CDF: Correlation

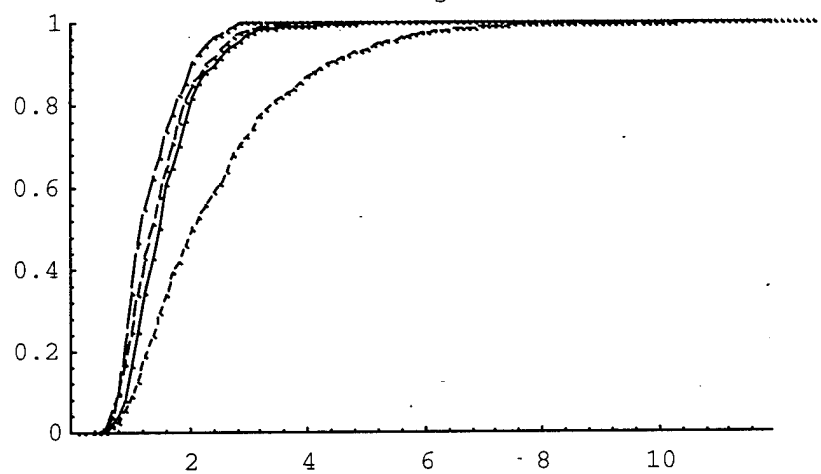




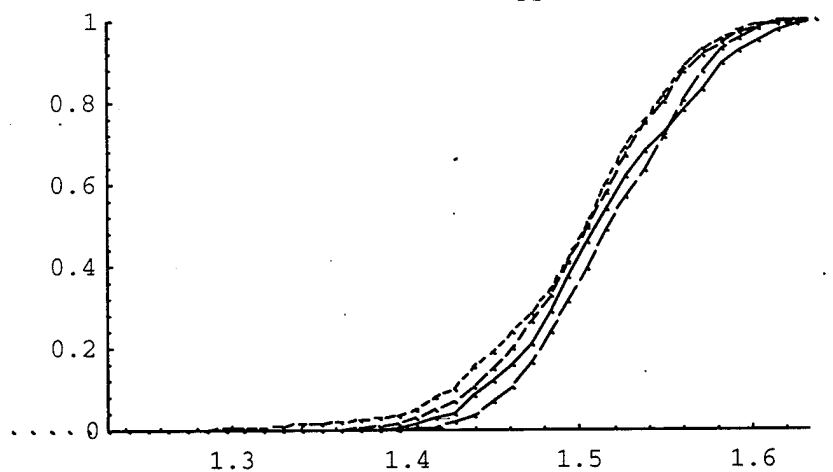
CDF: DiagonalMoment

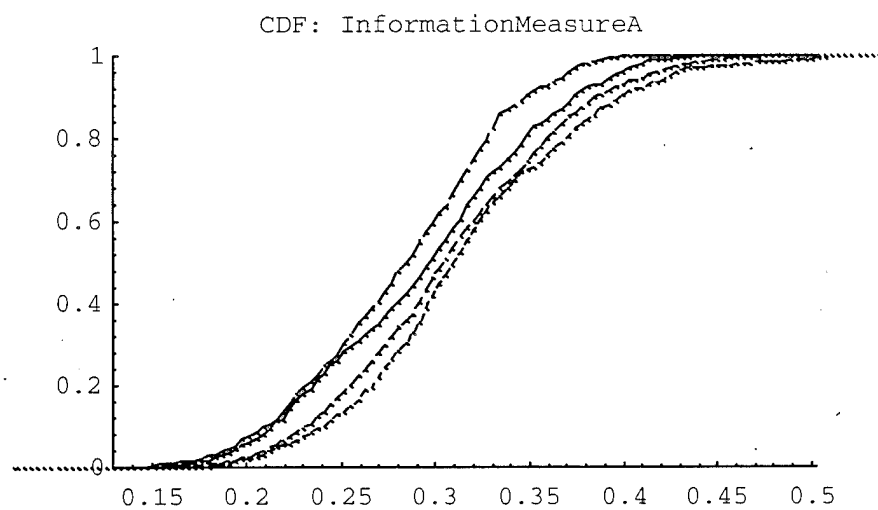
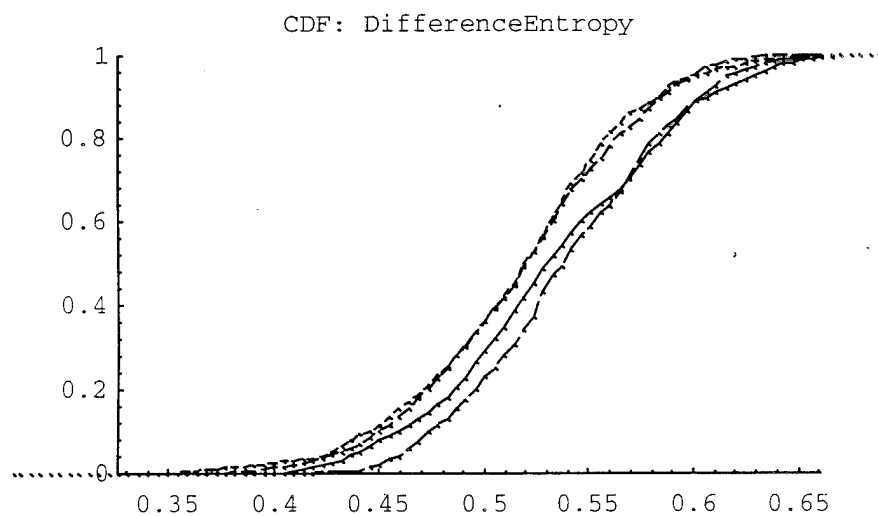


CDF: SecondDiagonalMoment

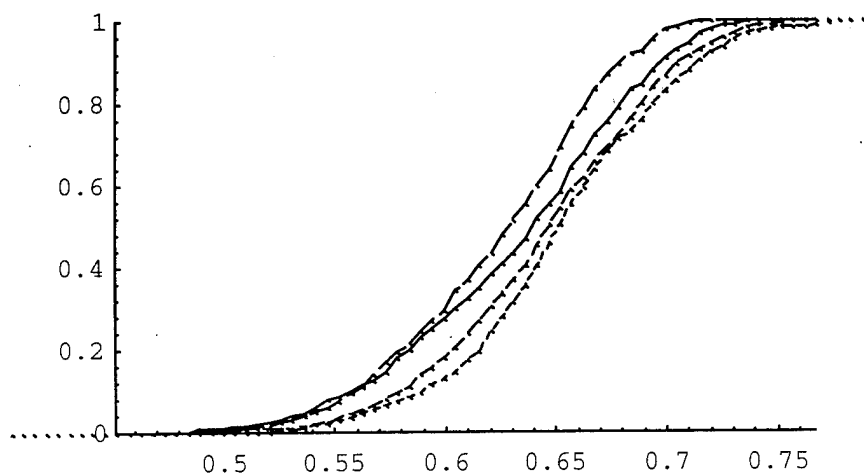


CDF: Entropy

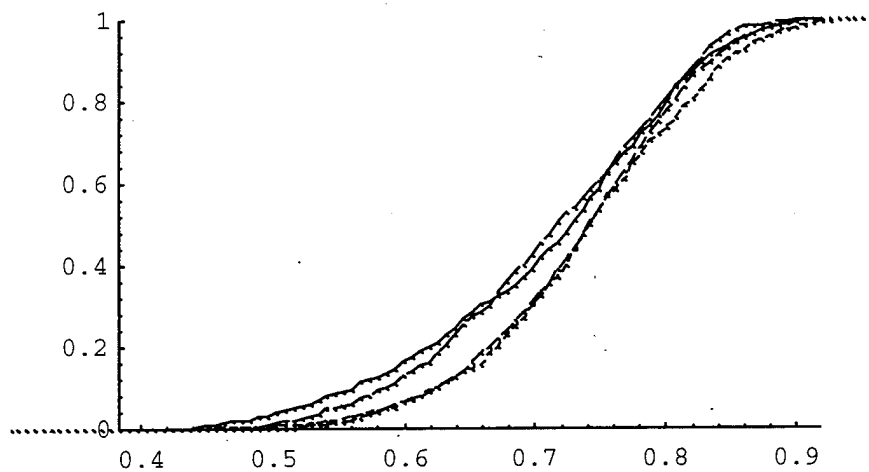


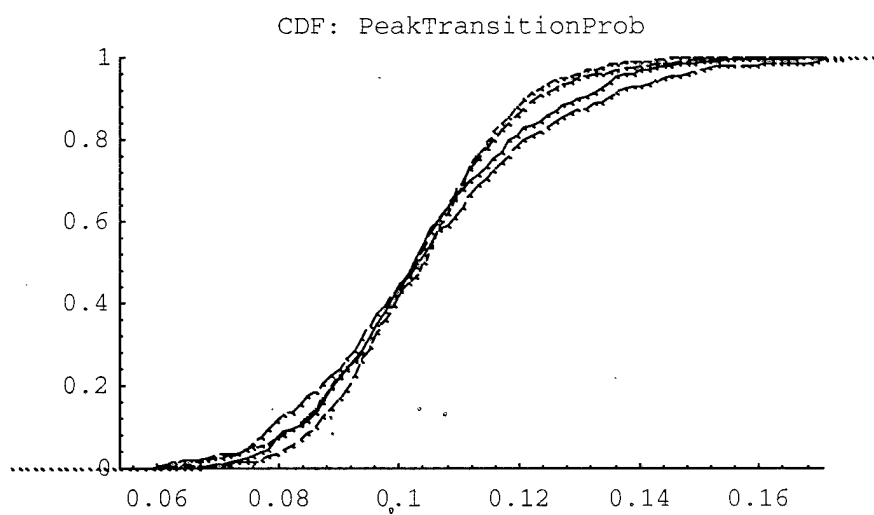
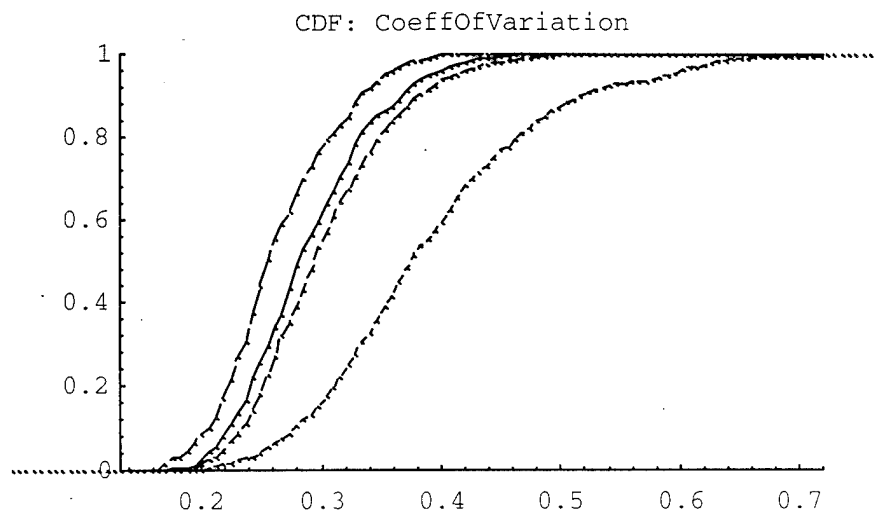


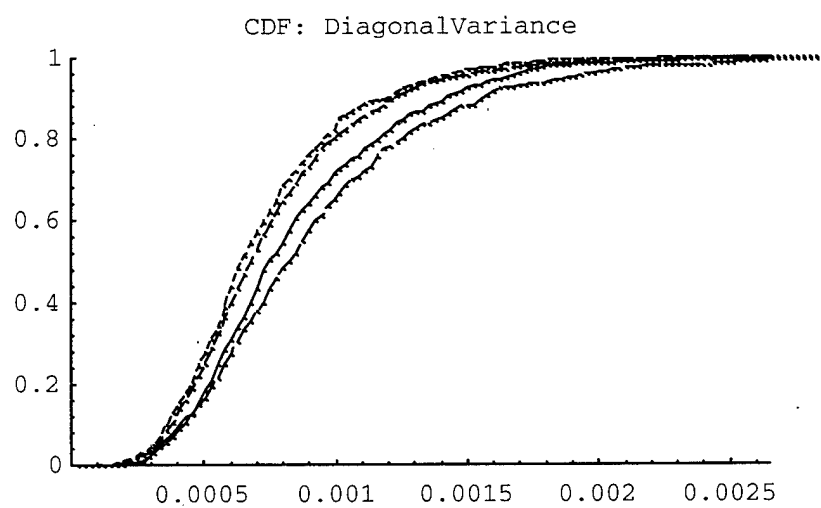
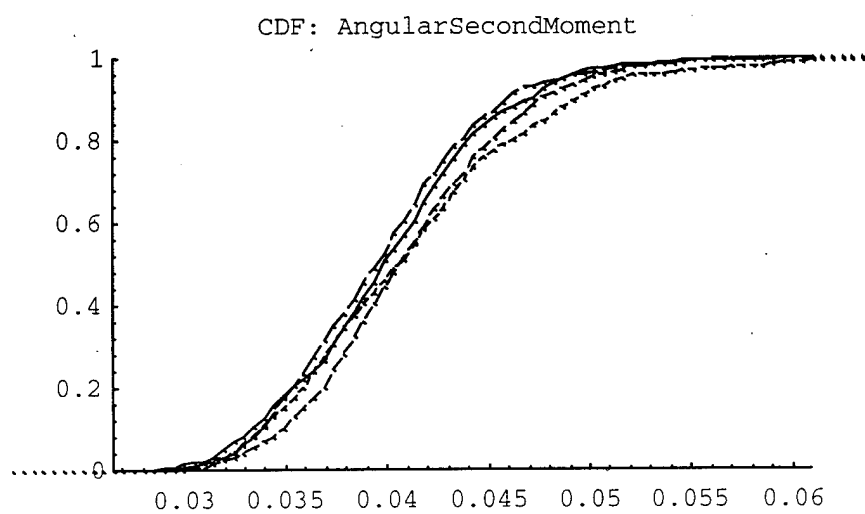
CDF: InformationMeasureB



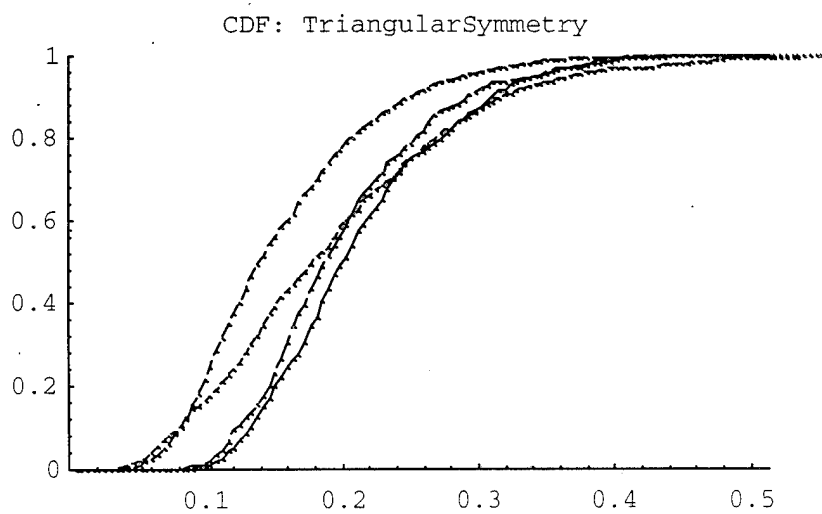
CDF: MaximalCorrelationCoeff







Triangular Symmetry *seems* to set the Normals apart from the rest of the pack:



II.3. FEATURE SELECTION

It is intuitively clear that many of the features are highly correlated. Cell Area is highly correlated with Perimeter, Min Diameter, Max Diameter, Cell Feret X, and Cell Feret Y. Elongation is highly correlated with Shape. Pg.DNA (i.e., DNA Content, measured in picograms) is directly correlated with DNA Index. And, the Summed Optical Density is directly correlated with DNA Index as well, because DNA content is computed directly from the optical density. On the other hand, Average Optical Density is correlated with both Pg.DNA as well as with Area, as it is directly correlated with Density, which is given by $\text{Pg.DNA}/\text{Area}$ (the difference from Density is that Average Optical Density is computed on a pixel-by-pixel basis).

By visually clustering the features, thereby eliminating obvious redundancies, we selected 11 features out of the 35 available. Many of the 35 features are highly correlated - this set of 11 are much less correlated, although it is likely that correlations remain among the subset of 11.

The reader should note that the set of 11 features presented here were selected from an analysis of 6 classes (Typical Normals, Atypical Normals, Benign, Cancer, Nondiagnostic, Lymphocytes), rather than the 4 classes considered here. The visual analysis shown here was performed after training runs (described in Part III) suggested that 4 classes might be more appropriate for this data set. Several of the 11 features selected according to the previous visual analysis appear to be informative for this set of 4 classes; however, some do not - in particular, some of the texture features could be substituted with others. As this fact was discerned towards the end of this preliminary study, we simply present the 11 features we used in Part III (which describes the training runs) and alert the reader that subsequent work should consider a reevaluation of the feature set.

Of the 11 selected features, the ones that measure morphometry, DNA content, or a combination thereof, are the following 5 features:

- | | |
|---------------------|---|
| 1. Area | Cell area (square microns) |
| 2. Shape | Cell perimeter divided by cell area. |
| 3. DNA Index | A normal A cell has an index of 1.0 when not dividing, and 2.0 when dividing. |
| 4. Density | Cell mass (measured in picograms of DNA) divided by cell area. |
| 5. Minimum Diameter | (microns) |

There are 12 morphometric and DNA features in all. Here are the remaining morphometric features:

- | | |
|---------------------------|--|
| • Perimeter | Perimeter of the cell border (microns). |
| • Summed Optical Density | Sum of optical density over all pixels (measured by gray scale). |
| • Average Optical Density | Summed Optical Density divided by number of pixels. |
| • Cell Feret X | Width of a bounding rectangular box around the object (microns). |

- Cell Feret Y Height of a bounding rectangular box around the object (microns).
- Maximum Diameter (microns)
- Elongation Maximum diameter divided by the minimum diameter.
- Pg. DNA Total content of cell DNA (picograms).

The texture features are Markov texture features, computed from a 8 by 8 gray level transition matrix. For detailed information on how the texture features are computed, see [Pressman 76, Haralick Shanmugam Dinstein 73].

There are 13 texture features. It is important to note that the early phase of this project explored the possibility of separating between 6 classes (Typical Normals, Atypical Normals, Benign, Cancer, Nondiagnostic, Lymphocytes), rather than just the 4 considered here (Normal, Benign, Cancer, Nondiagnostic). Therefore, the texture features that were selected were chosen because they appeared to provide separation that is not as apparent for the 4 classes shown here. Therefore, a continuation of this project would possibly benefit from a different set of features. See also the discussion in the section below titled "Preliminary Conclusions."

The 5 that were selected are the following:

6. Entropy
7. Information Measure A
8. Coefficient of Variation
9. Diagonal Variance
10. Inverse Difference Moment

The final feature selected can be considered a hybrid feature:

11. Standard deviation Standard deviation of the optical density of each pixel that composes the cell object.

Reasons for Selecting Each Texture Feature

The usefulness of the texture features is typically less obvious than that of the morphometric features. Here are the reasons each texture feature was selected:

1. Triangular Symmetry
Good indicator for Cancer and NonDiagnostic.
2. Coefficient of Variation
Good for NonDiagnostic, fair for Benign.

3 . Inverse Difference Moment

Fair to Good for Normals, fair for NonDiagnostic.

4 . Information Measure A/B

Fair for Normals.

5 . Entropy

This analysis was done on a lower level, splitting apart the Normal class into Atypical Normals and Typical Normals - under that analysis it appeared that Entropy was a fair indicator for Atypical Normals.

II.4. PRELIMINARY CONCLUSIONS

We evaluated the "informativeness" of features in terms of their ability to (individually) offer clear separation of one or more classes. For instance, it is clear that objects with very large values for Area are not Benign or Normal. However, the separation provided by most of the features is not as clear as that provided by Area. Also, even the separation provided by Area results in ambiguities at certain values of Area which can be exhibited by any of the 4 classes.

We can draw several positive conclusions from this visual analysis:

- 1 . The Nondiagnostic class appears to have many good indicators (Area, Shape, Min Diameter).
- 2 . The Cancer class has several good indicators (Area, DNA Index).
- 3 . The texture features can be expected to provide additional separation, given sufficient data. This is especially encouraging because texture is an aspect of the cell image that is difficult to discern subjectively by visual analysis alone.

Several less encouraging conclusions may be drawn as well:

- 1 . Many of the features provide little separation on their own (for this sample size). However, it is still possible that they might provide good separation when combined with other features.
- 2 . Where features provide class separation, they do so only *on average*. Therefore, it may require a large set of cells to utilize some of these features. Even the features that provide clear separation (e.g., Area, DNA Index) have regions of ambiguity inhabited by cells of all classes.

3. The Normal and Benign classes appear very much alike. Therefore, it may be difficult to distinguish Normal cells from Benign cells.

IMPROVEMENTS FOR SUBSEQUENT WORK

Avenues for improvement upon this preliminary feature selection are apparent. Note that the feature selection used to obtain the set of 11 features listed above was performed using 6 classes (the results of this visual analysis are not presented here, for clarity). Subsequent training runs (described in Part III) made it clear that, given the amount of data available, successful separation of classes would be more likely given fewer classes. This is also indicated by the results of the visual analysis presented here, (which was performed after the training runs, just prior to the writing of this report), which indicates that different features may be more useful for this set of 4 classes.

Also, while we selected a set of features that seemed to be informative with respect to providing separation among a set of four classes, we could also take a different approach: select a set of features for each class, such that the each set is most informative with respect to separating the corresponding class from the others. This is likely to be especially advantageous with respect to the Nondiagnostic class; further evidence of this is provided by the results of training runs (discussed in Part III). Subsequent work might use one set of features to cull out the Nondiagnostic cells, and then another set of features to distinguish among the remaining set of 3 classes.

II.5. REFERENCES

N.J. Pressman, "Markovian Analysis of Cervical Cell Images," J. Histochem. Cytochem. **24**, 138 (1976).

Haralick, R.M., K. Shanmugam, and I. Dinstein. "Textural Features for Image Classification." IEEE Transactions on Systems, Man, and Cybernetics. Vol. SMC-3, no. 6, November 1973.

AUTOMATING BREAST CANCER DETECTION**BY****NEURAL NETWORK CELL ANALYSIS****PART III: EXPLORATORY TRAINING RUNS****III.1. OVERVIEW**

Part III describes results of exploratory training runs performed on the breast cancer data. We conclude with an assessment of the avenues for continued research.

Due to the short amount of time under which this project was performed, the results ask more questions than they (conclusively) answer. A variety of different experiments were attempted, intended to explore the difficulty of using this dataset for training. We used rather conventional neural network training methods (conjugate gradient descent learning combined with multistart cross-validation, optimizing a squared error learning criterion) therefore, it is possible that these results could be dramatically improved by use of methods which may be more ideally suited for this type of classification task (however, we are in the process of applying a novel implementation of spline models to the task). It turned out that this dataset posed a challenging learning task to the methods we applied. We found strong evidence for several conclusions:

- 1 . The Nondiagnostic class seems to be most readily separated from the other classes.
- 2 . The Cancer class is the next easiest class to separate from the rest.
- 3 . The Normal and Benign classes seem to be easily confused.
- 4 . The Normal Atypicals and Normal Typicals were difficult to distinguish (within the data used here).
- 5 . It is likely that test case sizes will need to be increased to ensure an accurate diagnosis.

We also explored the use of Estimated Conditional Variance networks for use in providing “estimated certainty factors.” Intuitively, these ECV networks seem to be providing useful functionality, and we recommend further exploration of their use. However, it is clear that the major issues are getting better separation of training data, and determining whether it is clinically feasible to increase the number of cells in a tissue enough to result in an estimate that is clinically useful in real-world application.

III.2. THE LEARNING TASK

The objective here is to mimic the performance of a single human expert (here, Dr. Siderits) with respect to classifying cells into a set of classes from image data. It is also important that the model be easily applied to a typical case in a clinical setting - preferably, by culminating in a quantitative (in particular, probabilistic) assessment of a case as either Normal, Cancer, or Benign, along with the estimated certainty of that estimate. Therefore, an appropriately trained neural network will provide the Estimated Conditional Probability (ECP) of class membership given an example derived from a particular cell image.

To this end, we trained neural network models to estimate the probability that a cell will be classified in a particular class by our human expert. Theoretically, the output of a well-trained model averaged over a large enough test case should correspond to the class proportions of the test case. Here, we investigate the empirical validity of this on clinical data. We do this in steps:

1. Analyze the performance of neural net models with respect to their ability to successfully separate training sets generated from the cell image data used here.
2. Evaluate the usefulness of “Estimated Conditional Variance” (ECV) networks, which we use to learn a kind of “certainty factor” to be associated with the ECP networks.

3. Validate the ability of a well-trained model to automatically diagnose a test case.

Successful separation of the training data is a necessary, but not sufficient condition for successful training - the final proof is in whether the model can correctly diagnose a typical test case.

Because we deal here with small, finite, and inherently noisy datasets, we necessarily deal with additional uncertainty, even in the case that the trained model is unbiased (i.e., accurate, on average) with respect to a sizable training set. This is because variance is introduced from two sources: by randomness in the training data, and due to variance in the actual test cases. We can combat the former by using as much data as is available for use in training. However, to correct the latter, if necessary, will require modification of current clinical practice: rather than obtaining on the order of a few hundred cells per case, instead, obtain several thousand (or more) cells per case.

III.3. DATA

This section describes the data we used in training and testing. A preliminary analysis of the data prompted us to reserve as much of it as possible for use in training, with a few cases reserved for testing.

TRAINING DATA

We obtained the training data in sets of cases, where each case corresponds to an individual tissue sample, comprised of from over a hundred up to several hundred cell images. Here is the order in which we obtained the training data:

- 19 preliminary cases (well balanced among 6 classes: Typical Normals, Atypical Normals, Benigns, Cancer, Nondiagnostic, and Lymphocytes)
- 34 additional (mostly Typical Normals, Atypical Normals, and Benigns)
- 37 subsequent (mostly Cancer)

This is a total of 90 cases, for a total cell count of 25440 cells. We performed several training runs after each phase of data-gathering. The results of these exploratory runs were used to guide the experiments on the final set of 25440 cells.

Breakdown by Class Frequency

Here are the class frequencies for the final set:

- 7280 Normal (3163 Typical + 4117 Atypical)
- 6292 Benign
- 6446 Carcinoma
- 5422 Nondiagnostic (Garbage, Neutrophils, and Lymphocytes).

Analysis of CAS Autoclassification Performance

We had the CAS filter perform autoclassification on this dataset according to the filter settings (set according to the settings specified in Part I). The class frequencies assigned by the CAS are:

- 14945 Normal (1294 Typical + 13651 Atypical)
- 0 Benign
- 9945 Carcinoma
- 550 Nondiagnostic (Garbage, Neutrophils, Lymphocytes)

Of these, Dr. Siderits was responsible for changing 19601 of classification labels. Therefore, only 5839 of the CAS labels were correct.

Analysis of Total Time Required for ReClassification by Human Expert

There are 90 cases in this training set. Each case required from 10-15 minutes for Dr. Siderits to process and assign labels, plus several minutes afterwards to interpret the results. (Therefore, this database represents from 15 to 23 hours of analysis.) This is a "best-case" scenario, as this doesn't include additional time typically spent manually preprocessing the data (e.g., separating clumps in order to increase the number of usable cells, iteratively tweaking the CAS filters to deal with outliers in the dataset or to better deal with samples which are heavily populated with a particular class of cells). By comparison, a subjective analysis of an entire slide by visual inspection requires on the order of a few minutes.

TESTING DATA

We reserved 3 cases for use in validating the performance of the trained networks (more test cases are forthcoming).

- Mostly Benign: 493 cells, 335 of which are not Nondiagnostic.
- Mostly Normal: 493 cells, 386 of which are not Nondiagnostic.
- Mostly Cancer: 281 cells, 183 of which are not Nondiagnostic.

We note here the number of cells that are not Nondiagnostic to indicate how many cells of diagnostic import remain in typical test cases.

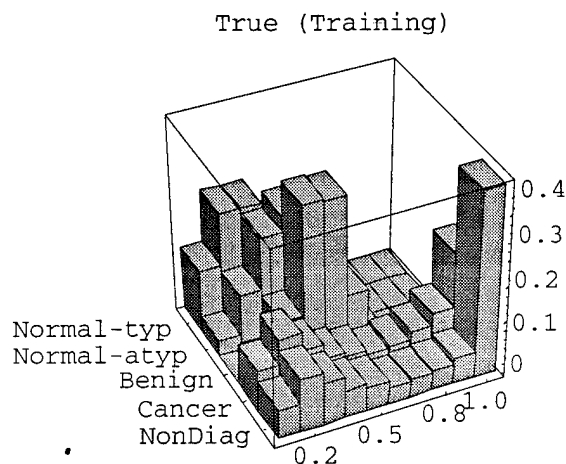
III.4. TRAINING RESULTS.

We explored a large number of different approaches, in terms of different classes, types of models, and methods of combining models. We generated over 1160 training individual training runs. This number includes only files we saved as being useful for subsequent analysis, and therefore omits a comparable number of training runs that were deleted or overwritten due to dead ends or lessons learned along the way. A substantial portion of these runs correspond to training several to many different networks, due to our use of multistart techniques in conjunction with multifold cross-validation. Therefore, suffice it to say that we will not attempt to catalog every training result we have obtained. Rather, we will do our best to digest the results and present our interpretation of the combined results using our best judgment, presenting quantitative evidence whenever this is possible without digression into unnecessary and distracting amounts of detail. Whenever possible, we illustrate the results of training graphically, to allow the reader to visualize the results (and in many cases, to draw their own conclusions, rather than summarizing results into a single quantitative score that can hide a great deal of interesting information).

ABILITY TO SEPARATE THE TRAINING DATA

The ultimate objective is to train models that generalize well to novel data. However, the presence of substantial amounts of noise in the training data requires us to first determine whether or not our models can successfully separate even large amounts of training data, on average. And, if so, what is the variance of the trained models? The analysis of Part II indicates that this may be an important issue: for example, is it possible to separate Atypical Normals from Typical Normals given this set of 25440 cells?

We toyed with using all 6 available classes in early experiments, but quickly determined that there was an insufficient number of Lymphocytes in the data set. We combined the Lymphocyte data with the Nondiagnostic data, and experimented with training neural networks to separate the cells into 5 classes. These results are best summarized by the following figure.



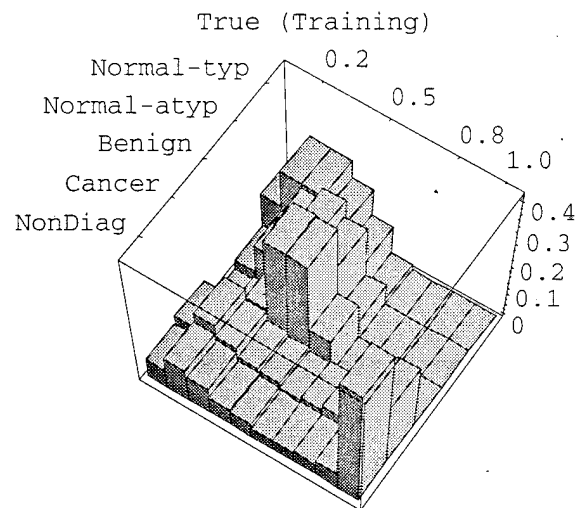
This 3-dimensional histogram shows the distribution of the 5 trained network outputs over the training set.

(The network outputs can be either from separate models or from a single model. We performed the experiment twice, once using 5 separate single output networks, and once using one network with five outputs. Both experiments yielded similar results, so far as the information depicted in this figure is concerned.)

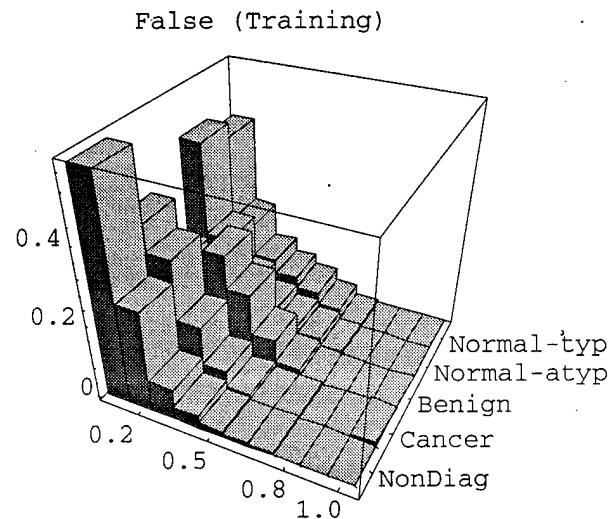
Each of the five histograms shows the distribution of a particular output upon *in-class data*. The horizontal axis measures the ECP given by the network output binned into 10 bins given by the intervals $[0,0.1)$, $[0.1,0.2)$, ..., $[0.9,1.0]$. The vertical axis gives the probability of the network output falling into a given bin. To obtain the histogram for the Normal-Typical output, the network is provided with only training examples which are labeled "Normal-Typical," and the value of the Normal-Typical output is binned accordingly. This procedure is repeated for each class. Therefore, the optimal separation for the "Normal-Typical" class would occur if all of the in-class examples for that class fell into the $[0.9,1.0]$ bin.

Indeed, the Normal-Typical, Normal-Atypical, and Benign classes did not fare well under this analysis. However, it is encouraging that the Cancer and Nondiagnostic classes gave a histogram which, although not optimal, had the desired shape.

It is somewhat difficult to see all of the data in this figure, so we provide another, more overhead view of it here:

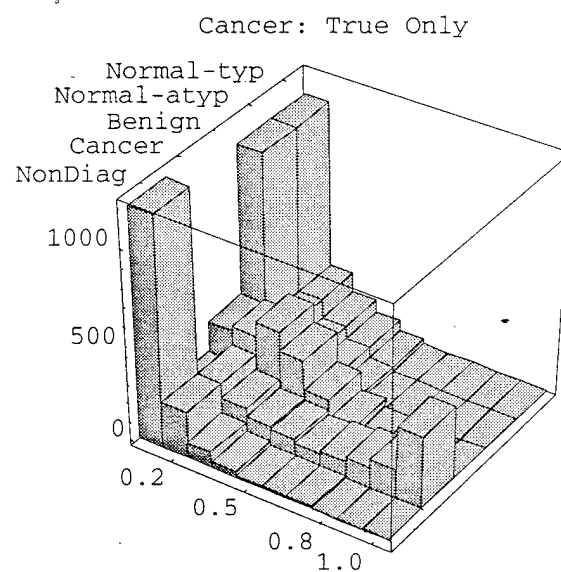
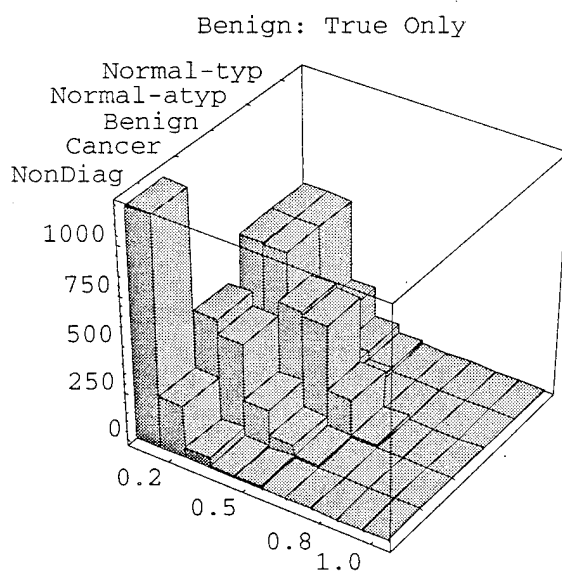
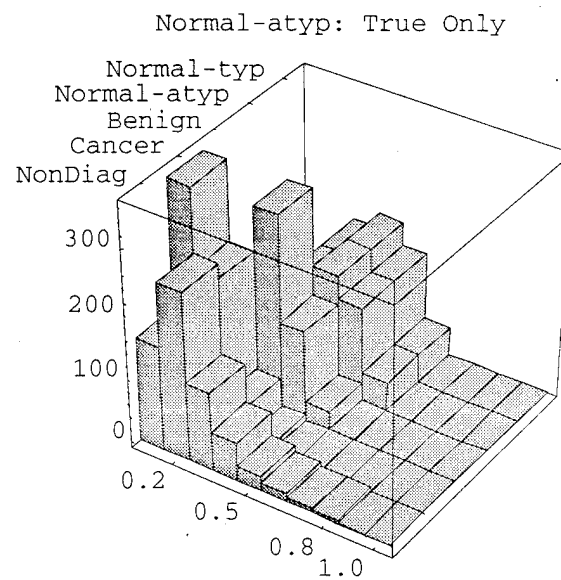
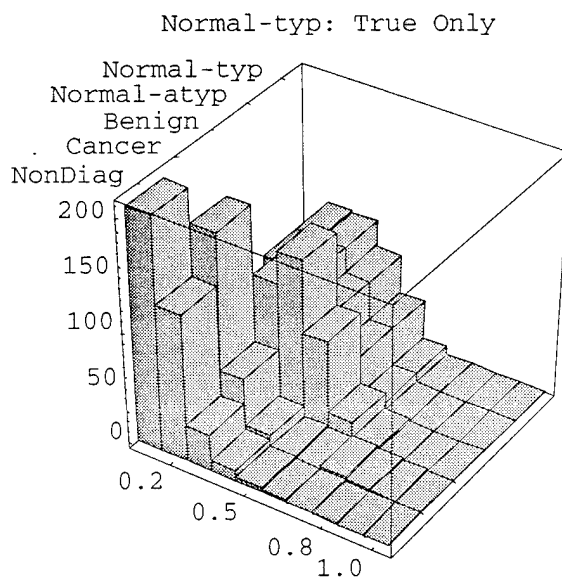


The corollary to this analysis is given by creating the same figure, but instead, looking at the distribution of each output on out-of-class data. This is given in the following figure:



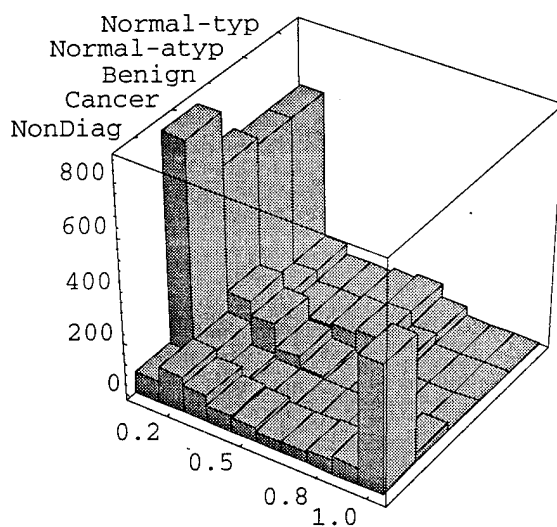
Note here that each of the individual distributions is qualitatively close to the desired shape, except for the Benign class. Therefore, for the most part, each trained "expert" (other than Benign) does fairly well at discerning that a cell is *not* in its class.

Note that the Benign trained "expert" gives an ECP distribution over out-of-class data in a manner similar to its ECP distribution over in-class data (see the previous figures above). From this evidence, we suspect that this network is confusing the Benign class with other classes, and vice versa. We can home in on the cause for this behavior by performing another analysis of the network outputs. In this analysis, we give the network examples from one class, and then plot the distribution of the network output over all of the outputs. This illustrates the propensity for the network to confuse an example for the incorrect class alongside its ability to associate it with the correct class, as evidenced by the distributions of the ECPs over each of the network outputs. The top two figures show the distribution of the network outputs upon Normal-Typical and Normal-Atypical training examples only, respectively, and the bottom two show the same thing for the Benign and Cancer classes.



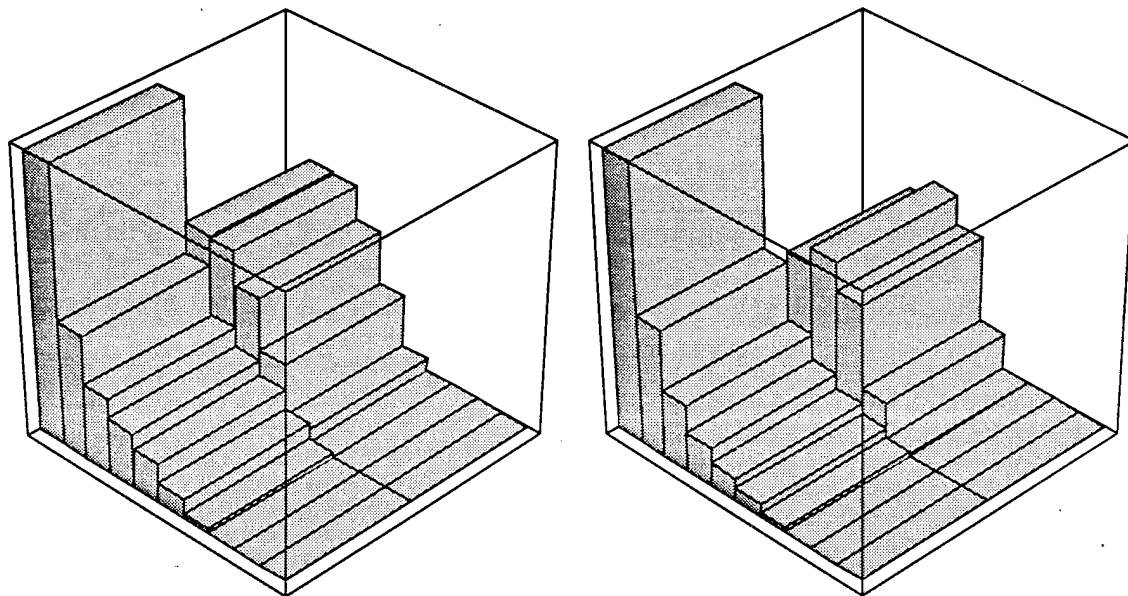
Note that the Normal-Typical and Normal-Atypical classes seem to be confused with each other (as their distributions are almost identical across each of the figures). The Benign network output appears to provide no separation at all, becoming undesirably strong for examples of other classes, especially for the Normal-Typical class. Indeed, the Benign "trained expert" appears to give the same output for all examples, on average. On the other hand, the "Cancer" expert seems to be giving good separation, for both in-class and out-of-class examples.

The following figure shows the distribution of the network outputs on Nondiagnostic in-class examples.

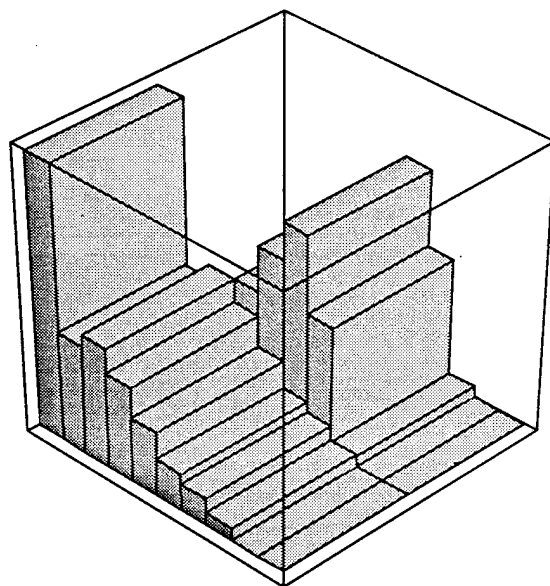


It is apparent that this model is good at separating the Nondiagnostic and Cancer classes, and does less well at separating the Normal and Benign classes.

Now we address the issue of the apparent confusion by the model between the two Normal classes. We can improve the situation somewhat by clumping the Normal Typical and Normal Atypicals into a single class called "Normals." The following two figures separate out the Normal Typical (on the left) and the Normal Atypicals (on the right). Each figure shows the performance of the corresponding trained network output on both the out-of-class examples (given by the foreground histogram) and in-class examples (given by the background histogram). Ideally, the foreground histogram would have all bins empty except for the far left bin, and conversely, the background histogram would have all bins empty except for the one on the far right. (Clearly this is not so.)



On the other hand, if we combine these two classes together, we get the following separation over the training set:



This is clearly an improved, although still not ideal distribution. (These figures correspond to a four single-output neural network; we obtained similar results for a single four-output neural network. We also obtained similar results for a network trained using a smaller training set using the first 5 features.)

Increasing the flexibility of the model did seem to improve separation somewhat, but not enough to make this effect disappear. This suggests several additional avenues for additional research to pinpoint the causes for this apparent inadequacy:

- 1 . Evaluation of methods for improving the feature selection process.
- 2 . Evaluation of different models and training methods.
- 3 . Evaluation of the amount of noise that is inherent to the data (e.g., would the same doctor reclassify the same data the same way on a different day? How would a different doctor classify the same data?).

However, it is important to note that the network shown here is accurate, on average, over the entire training set.

The relative frequencies of the classes in the training set are

- 28.6% Normals,
- 24.7% Benigns,
- 25.3% Cancer,
- 21.3% Nondiagnostic.

This network estimate the class frequencies to be

- 28.9% Normals,

- 26.3% Benigns,
- 25.0% Cancer,
- 22.6% Nondiagnostic.

(These averages sum to 1.03.)

However, it is clear from the figures that, while the ECPs are accurate on average over the entire training set, the trained experts for the Normal and Benign classes are clearly more biased (and probably have more variance as well) than the Cancer and Nondiagnostic classes. It is of interest whether we can detect this effect independently. This is explored in the next section.

ECV NETWORKS

We trained a 4-output ECV network to estimate the squared error of the ECP model (again, we obtained similar results for 4 single-output neural networks). Each output of this network is trained to estimate the residual error of the corresponding output of the ECP network. Here are the “Estimated Conditional Variances” given by the ECV network, averaged over the entire training set:

- Normals: 0.155
- Benigns: 0.150
- Cancer: 0.118
- Nondiagnosics: 0.068

A lower ECV is better. Note that these are estimates of average squared error. Therefore, a model that makes no prediction at all but which simply gives a constant output of 0.50 would have an average error of 0.5, corresponding to a squared error of 0.25 (therefore, an ECV of 0.25 indicates that the corresponding expert is doing no better than

random guessing; more than 0.25 indicates that it is worse). Here's the same list, but with the square root of the ECVs instead:

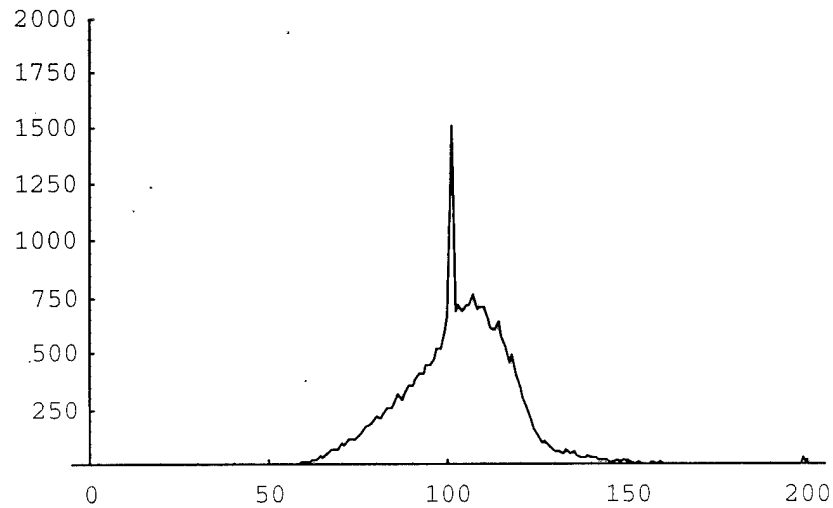
- Normals: 0.394
- Benigns: 0.388
- Cancer: 0.343
- Nondiagnosics: 0.260

This matches our intuition that the models we've seen do better at separating the Cancer and Nondiagnostic classes from the rest than they do at separating Normals or Benigns. The fact that the ECVs for the Normal and Benign classes are so much higher relative to the Cancer and Nondiagnostic classes serves as a warning as to the performance of this model on test data.

We were not able to investigate as deeply as we would have liked into the relevance of the ECV networks with respect to detecting highly uncertain predictions; we suggest this as an important avenue for further research.

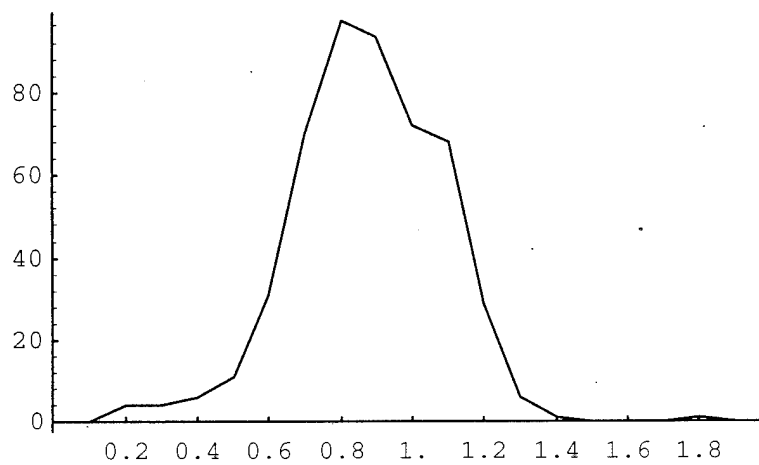
DISTRIBUTION OF THE SUMMED ECPS

We evaluated another possible indicator of model certainty given by the sum over the individual trained experts. Note that the training procedure used here uses targets in $\{0,1\}$, yielding a set of trained experts that are interpreted as ECPs. Therefore, ideally, the ECPs for each of the 4 trained experts should add to 1.0. However, they are not constrained to do so under this training method (as compared to Bridle's Softmax method). Therefore, we evaluated the utility of the sum of the individual ECPs as an indicator of the model's reliability. The model depicted above generated the following distribution of summed ECPs over the training set, where 100 corresponds to 1.0:

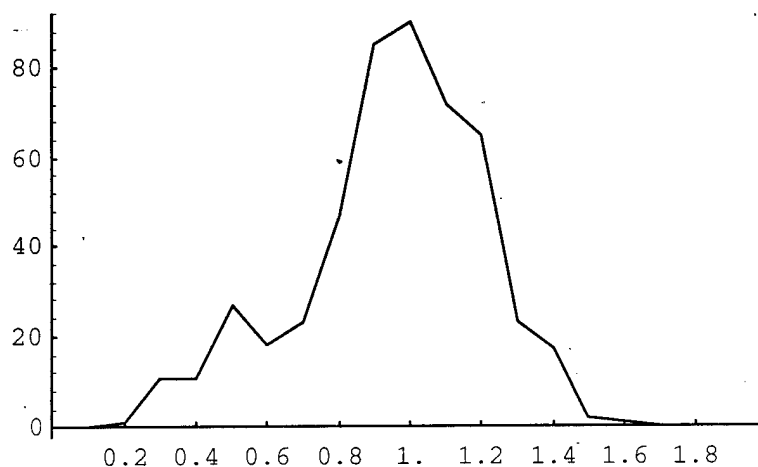


Here is the same type of graph (for a 3-output network trained on 3 classes: Normal+Benign, Cancer, and Nondiagnostic) evaluated over the 3 test cases (recall that these test cases are comprised of a few hundred cells).

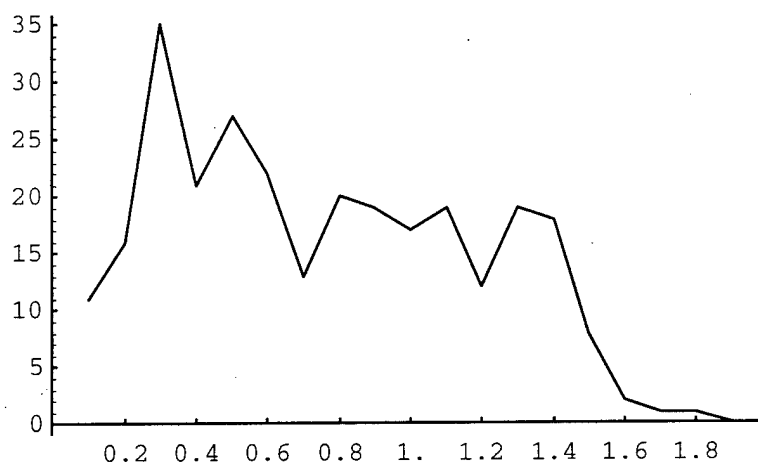
The first figure corresponds to the test case comprised of mostly normal cells:



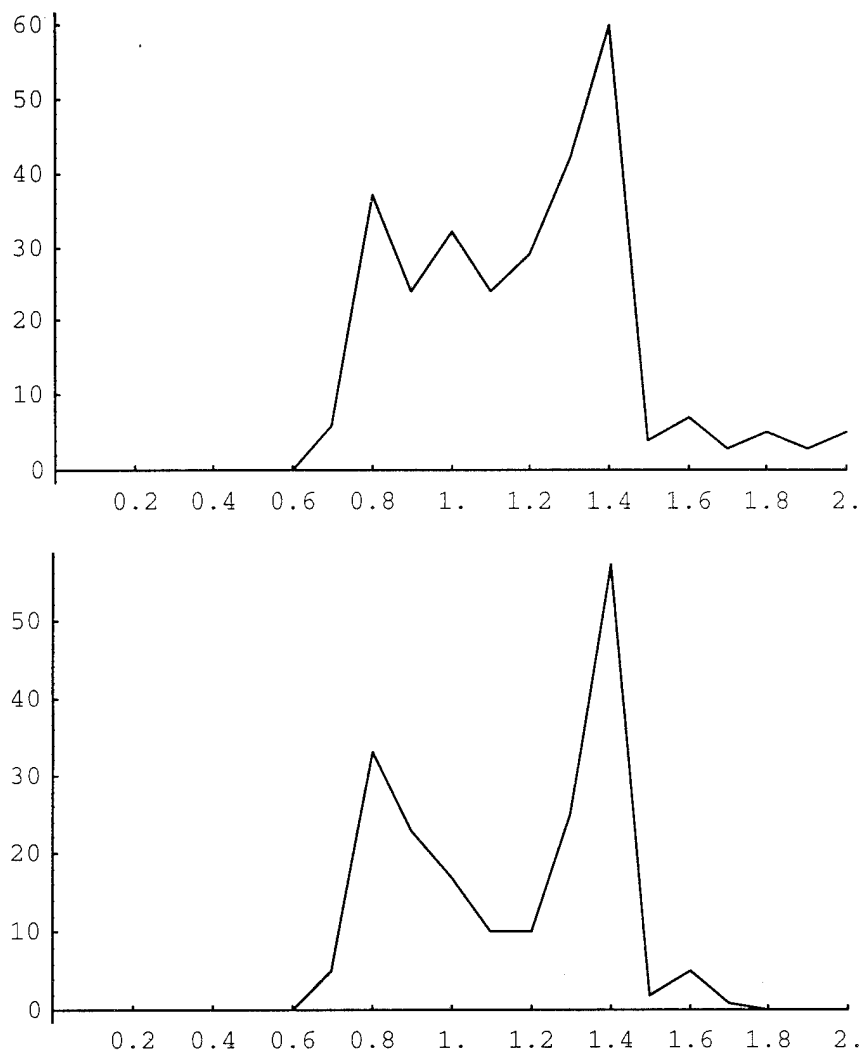
The next one corresponds to a test comprised of mostly benign cells:



The next one corresponds to a test case comprised of mostly cancer cells:



We noticed the same effect in two 4-output networks (trained on classes Normal, Benign, Cancer, and Nondiagnostic), as shown in the following two figures:



It is difficult and unwise to extrapolate from a few networks evaluated on 3 test cases; however, these figures suggest further exploration into the usefulness of the Summed ECPs as an additional indicator of cell class.

DIAGNOSTIC PERFORMANCE ON TEST CASES

The analysis so far warns that the models being evaluated here may have problems on test data with respect to correctly identifying some of the classes, due to the less than desirable performance on training data. This suspicion is corroborated by experiment 4-output models performed poorly with respect to correctly diagnosing the 3 test

cases for the correct proportion of the Normal, Benign, and Cancer classes, but were fairly accurate with respect to the Nondiagnostic class. Therefore, one of our objectives (that of being able to automatically filter Nondiagnostic objects) appears to be technically feasible.

Here are results of detecting the Nondiagnostic Class from 3 test cases. All values are percentages. The "Actual" row gives the actual percentage of the case which is comprised of Nondiagnosics. The next row gives the ECP estimate of this percentage. The WTA row gives the winner-takes-all estimate of this percentage, with the last row giving the percentage of examples classified as Nondiagnostic by WTA that actually were Nondiagnostic (i.e., the "True Positives").

Cases:	Normal	Benign	Cancer
Actual percentage:	22	35	32
ECP estimate:	24	29	30
WTA estimate:	14	19	19
%TP:	88	92	69

The success of the networks with respect to being able to identify the Nondiagnostic class suggested an additional approach to improving the overall classification performance. This method culls out the Nondiagnosics, and then proceeds to classify the remaining classes using a set of features that are more appropriate.

We created a training set with only Normal, Benign, and Cancer examples, having no Nondiagnostic examples. We then selected a set of 7 features that appear to be relevant to separating these 3 classes (thereby eliminating several features that are good for the Nondiagnostic class, but less informative for the other 3 classes).

Here are the 7 features:

1. Area

2. DNA Index
3. Density
4. Standard Deviation
5. Correlation
6. Coefficient of Variation
7. Triangular Symmetry

We then trained several networks on this set of features (over the training set of 20018 examples obtained by filtering out the Nondiagnostics from the original training set of 25440 examples), selecting the best one by cross-validation. The resulting network did perform somewhat better on the set of 3 test cases. The following table gives the results of testing this network on 3 test cases.

	Normal test case:	Benign test case:	Cancer test case:
TOTAL ERROR:			
Mean Squared Error (MSE):	0.194	0.243	0.190
Root MSE:	0.440	0.494	0.436

That the RMSE is not much less than 0.5 (the RMSE that can be obtained, for example, by random guessing) indicates that these results are less than desirable.

The following tables breakdown the performance of this model according to the 3 "trained experts," giving the actual proportion in the test case followed by two estimates derived from the network output. The ECP is given by the average output of the trained expert over the test set. The "WTA Average" is a winner-take-all method to classifying each individual cell as either in-class or out-of-class (rather than ascribing a real-valued probability of class membership), and is described in the next subsection.

Average WTA

One way to combat the effects of small finite test sets is to require the trained model to do less than what it was trained to do. One way to apply this concept to our problem is to train 4 models, one for each class (or alternatively, a single model with 4 outputs, one for each class), and then instead of using the individual network outputs to estimate probability of class membership, use them to obtain a binary consensus of the class membership for each cell by giving a value of 1.0 to the maximal network output, and 0 to the others. We refer to this method of arriving at a consensus as "Winner Take All." To obtain the diagnosis over a test case, average the WTA consensus for each class over the entire test case.

Testing on a Clinical Test Data

We tested this network on 3 clinical test cases, one each for Normal, Benign, and Cancer, respectively. The Normal case had 386 cells, comprised entirely of Normal cells. The Benign case had 335 cells, comprised entirely of Benign cells. The Cancer case had 183 cells, comprised of mostly cancer cells (96% Cancer, 4% Normals).

The following figure tabulates the results. The columns correspond to the 3 test cases, the 3 sets of rows correspond to each of the 3 trained experts. All values are percentages. Where applicable, the WTA average is supplemented by a the percentage of True Positives (in parentheses).

	Normal test case:	Benign test case:	Cancer test case:
NORMAL:			
Actual % :	100	0	4.4
ECP:	39	42	33
WTA average:	77 (100)	93	38 (12)

BENIGN:

Actual % :	0	100	0
ECP:	32	32	33
WTA average:	0	0	0

CANCER:

Actual % :	0	0	96
ECP:	32	30	40
WTA average:	23	7	62 (100)

Analysis

It appears that this model is simply unable to separate the Benign class from Cancer or Normals. The model has essentially "turned off" the Benign expert by setting it to a constant function equal to approximately 0.32 (corresponding to the proportion of Benigns in the training set: 6446/20018). This effect was repeatable: it was also observed with the 4-output network, where the Benign output was approximately constant at about 0.25 (approximately equal to 6446/25440, which was the ratio of Benign in-class examples to the total training set size in that case).

The model does much better at distinguishing between Cancer and Normals, with each the Normal and Cancer experts giving a stronger response to the appropriate in-class case. The Cancer and Normal test cases would have been diagnosed correctly in this case. Although this is true for both the Average ECP interpretation as well as the Average WTA interpretation, the WTA diagnosis gave a stronger indicator.

Note that the WTA classification was correct 100% of the time for the Normal and Cancer experts given a case dominated by the corresponding class.

TRAINING ON NORMALS AND CANCER EXAMPLES ONLY

We created a new training set by filtering the Normal and Cancer examples from the training set used above. This results in a set of 13726 examples. We trained a set of single-hidden layer networks, with from 1 to 5 hidden units, using multifold cross-validation to select the number of hidden units. The cross-validation procedure used 20 splits of the training set, each split using 95% of the data for training, and 5% for testing. This procedure selected a network with 2 hidden units.

Here are the overall error results for each of the 3 test cases:

	Normal test case:	Benign test case:	Cancer test case:
TOTAL ERROR:			
Mean Squared Error (MSE):	0.04	0.37	0.10
Root MSE:	0.20	0.61	0.32

And here is a breakdown by expert:

	Normal test case:	Benign test case:	Cancer test case:
NORMAL:			
Actual % :	100	0	4
ECP:	88	82	19
WTA average:	97 (100)	97	20 (22)
CANCER:			
Actual % :	0	0	96
ECP:	12	18	81
WTA average:	3	3	80 (100)

The Cancer expert gives a clear response to the cancer case, for both ECP and WTA diagnosis, whereas the Normal expert is relatively weak. Likewise, the Normal expert gives a clear response to the normal case, with respect to both WTA and ECP, with the Cancer expert giving a very weak response. Where applicable, the WTA Average is supplemented by a value (in parentheses) which gives the percentage of True Positives out of the cells classified as in-class by the WTA consensus method. Surprisingly, the True Positive rate for WTA consensus was 100% for both of the dominant classes (this doesn't apply to the Benign case as that case has no Normal or Cancer cells). Therefore, the error rate was due to False Negatives, and when the WTA estimate was large, it was also accurate, at least with respect to giving a lower bound (and hence, conservative) estimate of the in-class probability.

Interestingly, the Benign test case is diagnosed as "Normal." Of course, additional testing is required to conclude much from this (as well as all of these results), however, this is the desired behavior. This may be due to a bias in

this particular model to classify Benign cells more like Normals than as Cancer, however, the cumulative set of results we've seen here suggest that there may be a deeper similarity between Benign and Normal cells that is being detected by these models.

III.5. CONCLUSIONS

MAIN RESULTS:

The evidence here suggests that it is technically feasible to automatically filter Nondiagnostic cells from tissue samples.

While the methods we used had difficulty separating some of the classes when presented with all of them simultaneously, it was possible to dramatically improve upon the classification performance by using two-stage detection: one stage to detect and cull out Nondiagnosics, and another stage to discriminate between Normals and Cancer, given that there is no presence of Nondiagnosics.

It is likely that the size of the test cases would need to be increased significantly to use these methods in clinical application. We expect that this would provide a further improvement in the results.

The Estimated Conditional Variance training did seem to work well, intuitively. It indicated how the trained experts performed relative to each other, as well as individually.

AVENUES FOR CONTINUED WORK:

The most important goal would be to test the two-stage method (culling out Nondiagnosics followed by discriminating between Normals and Cancer) in clinical practice.

The general methods we evaluated here could be improved upon in several ways.

It is possible that our results could be dramatically improved by increasing the number of cells in the test cases, from a few hundred to a few thousand, or more, if necessary. We would strongly recommend this course of action, especially for the purpose of detecting Benign cells, as well as for discriminating between Cancer and Normals. This is especially important due to our finding that a two-stage technique may be successful in first culling out the Nondiagnosics and then performing two-class discrimination on the remaining cells, because every tissue sample typically contains from 20 to 30 percent Nondiagnosics, therefore culling these out automatically will cut the sample size down significantly, especially if the False Positive rate is allowed to be high to ensure total automation.

It would be desirable to improve on the separation of the training data to determine whether the Benign class can be detected. This might be attainable via more rigorous feature selection, application of different models and training methods, or optimization of different training criterion suited to classification learning tasks.

It might be useful to apply a mixture of experts, with a number of experts per class, where each expert uses a possibly different set of feature vectors.

It is possible to adjust the interpretation of a model that gives an uneven separation of classes to provide a usable classification, and it appears from the graphical depiction of our results that there is room for application of this approach to the training methods used here; however, we first recommend attempting to improve separation during training.

This project ended just as new data became available for testing. The amount that was available at the time of the writing of this report was just enough to illustrate our results and provide a glimpse of whether an approach was feasible or on the right track. We were unable (with only 1 test case per class) to analyze whether the models were unbiased on average over a number of test cases, to test for a systematic bias.

It would be very interesting to use additional testing data to validate the use of the ECV models, to determine whether the ECV is correlated with an intuitive notion of certainty for real test cases (i.e., given small test cases for which we expect the variance of the prediction to be high, is it possible to detect when it can be expected to be especially high due to the composition of cells in a given tissue sample?).

III.6. REFERENCES

N.J. Pressman, "Markovian Analysis of Cervical Cell Images," J. Histochem. Cytochem. **24**, 138 (1976).

Haralick, R.M., K. Shanmugam, and I. Dinstein. "Textural Features for Image Classification." IEEE Transactions on Systems, Man, and Cybernetics. Vol. SMC-3, no. 6, November 1973.

Classification of Cancer Cell Data using Spline-Nets

KATrix Inc.
Stephen H. Lane
Kam Jim
July 14, 1995

Summary

Using a Spline-Net with 20 hidden-nodes and an adaptive exponential-decay model (EDM) Bifurcation Schedule, it was found that the CAS database consisting of 25,000 examples of cancerous, benign and normal cells could be classified with an average of 2.0% error in 200 epochs. However, it was found that there is a significant amount of data in and between the training and test sets with high correlated input patterns but conflicting output classifications. As a result, Spline-Net generalization was poor on the test set data, as would be expected. By removing the correlated patterns from the training and test sets, generalization on the test sets is improved.

Spline-Net Architecture and Training Procedure

A Spline-Net with 20 hidden nodes was used in all simulations reported here, and appears to be optimal for the experiments conducted. An adaptive bifurcation schedule based on an exponential-decay model also was used to bifurcate each the partitions in the connection functions of the hidden-layer and output nodes based upon the output node mean-square error.

Simulation Results

Five runs were performed for each simulation. The simulations made use of the following files provided:

database.trn	database of 25,440 cell examples
benign.tst	database of 493 patterns of mostly benign cell examples
cancer.tst	database of 281 patterns of mostly cancerous cell examples
normal.tst	database of 493 patterns of mostly normal cell examples

Learning the Individual Data Sets

The simulations conducted indicate that the Spline-Net is capable of classifying the patterns in the above data sets. These results are shown in the first row of Table 1.

The Spline-Net can learn to classify the individual test sets (benign.tst, cancer.tst, normal.tst) with 0% error. The Spline-Net also can learn to classify all three files together with an average of 6.3% error. A correlation measure ranging from [-1.0, 1.0] was then used to detect correlated input patterns. A correlation value of 1.0 means the data is colinear, a value of -1.0 means the data is anti-correlated. When all highly correlated input patterns (i.e., > 0.95 correlation) with conflicting output classifications are removed, the Spline-Net can learn to classify the data in the three files with 0% error.

Training was performed on a sub-set of the 25,440 example training set, called the *Working Set*, which consisted of all patterns in the training set whose output error on the previous training epoch was above a pre-specified threshold. Every 10 epochs all training patterns were placed back into the Working Set. Use of a Working Set dramatically reduced the amount of training time required by allowing the Spline-Net to spend more time learning the patterns that contributed significantly increased the output node mean-square error function. The test sets were used as validation sets during the training process to prevent the Spline-Net from over-fitting the training data. It was found that the Spline-Net was able to classify the cells in the 25,440 example database with an average of 2.0% error in 200 epochs.

Generalization

The generalization results are shown in the 2nd and 3rd rows of Table 1. After training on the 25,440 example database to 2.0% error, classification on the test sets benign.tst, cancer.tst, and normal.tst results in 72%, 19%, and 41% average classification errors, respectively. The overall generalization error is 48%. Generalization on the 3 files after all highly correlated (> 0.95) patterns with conflicting known classifications are removed, was 17%. The better generalization is expected, since the difficult classification patterns have been removed.

A modified training set was compiled consisting of the first 5,000 patterns of the 25,440 example database, but with all highly correlated (> 0.99) patterns having conflicting classifications removed. Assuming the removed patterns are inconsistencies or outliers, the new training set should improve the Spline-Net generalization performance. If the removed correlated patterns are not actually inconsistent, then additional input features (removed to simplify the problem) may have to be reintroduced to the input data in order to reduce the apparent classification conflict. The training set that results after removing the correlated patterns contains 4,643 examples. After 200 epochs, an average classification error of 1% was achieved on this training set. Generalization was slightly improved. Classification on the test sets benign.tst, cancer.tst, and normal.tst resulted in 69%, 33%, and 20% average classification errors, respectively. The overall generalization error was 42%. The above experiment was repeated, but with the correlation threshold relaxed to 0.95. The resulting training set contained 1,950 patterns. After 200 epochs, an average classification error of 0% was achieved on this training set. Generalization on the test sets benign.tst, cancer.tst, and normal.tst resulted in 66%, 30%, and 15% average classification errors, respectively. The overall generalization error in this case was 38%.

Generalization error on a modified test set consisting of the benign.tst, cancer.tst, and normal.tst data sets with all highly correlated (> 0.99) patterns of conflicting classification removed was 15%. When the correlation threshold was reduced to 0.95, generalization on this test set resulted in 10% classification error.

TRAINING		GENERALIZATION	
Training Set	Error	Test Set	Error
benign	0%		
cancer	0%		
normal	0%		
benign + cancer + normal	6.3%		
benign + cancer + normal (.95)	0%		
25K cell database		benign	72%
		cancer	19%
		normal	41%
		benign + cancer + normal	48%
		benign + cancer + normal (0.95)	17%
25K cell database (0.99)		benign	69%
		cancer	33%
		normal	20%
		benign + cancer + normal	42%
		benign + cancer + normal (0.95)	15%
25K cell database (0.95)		benign	66%
		cancer	30%
		normal	15%
		benign + cancer + normal	38%
		benign + cancer + normal (0.95)	10%

Table 1: Classification errors on various data sets. A number in parenthesis after a file name indicates that all patterns with cross-correlations above the threshold in parentheses are removed if they have conflicting classifications.

Future Work: A Brief Cluster Analysis of the Database

A brief analysis was performed to determine the number of clusters of correlated patterns in the database. All cell examples which have a similar correlation (above a pre-specified threshold) were grouped into the same data cluster. The analysis was applied to the first 12,000 examples in the 25K example training set. The results are shown below in Table 2.

Threshold Value	Number of clusters
0.95	1,490
0.90	315
0.85	116
0.80	45
0.75	23

Table 2: Number of clusters as a function of input pattern correlation threshold

The results in Table 2 suggest that a large fraction of the input data is fragmented into highly correlated groups of data clusters. To improve the generalization of the Spline-Net, future work would pre-process the training data into correlated groups, then train a separate Spline-Net to classify the data in each group. Test data (or new data from a clinical trial) would first be processed to determine the correlation of the input patterns with the centroids of the cluster groups, then classified by the Spline-Net with the highest associated centroid correlation. The resulting approach would allow cluster analysis to coarsely classify the input patterns while enabling the Spline-Net to focus on the fine details associated with correctly classifying the output data.

APPENDIX B

Technical papers included in quarterly reports submitted previously

Oct-Dec'94

1. De Vries, B., "Gamma Neural Networks for Word Spotting". Final Report delivered to Allen Reeves.
2. S.Y. Kung and J.S. Taur, "Decision-based Hierarchical Neural Networks with Signal/Image Classification Applications." IEEE Transactions on Neural Networks, Vol. 6, No. 1, pp. 170- 181, January 1995.
3. S.H. Lin and S.Y. Kung, "A New Approach to Sensor Fusion via Expectation-Maximization (EM) on Hierarchical DBNN", submitted to Inter. Conf. on Image Processing, Washington, D.C., 1995.

Oct'93-Sep'94 (annual)

1. Lane, S.E. and K. Jim, "Adaptive SPLINENET Training Schemes", September 19, 1994
2. S.A. Markel and C. Fefferman, "Recovering a Feed-Forward Net from its Output", SIAM 94 presentation.
3. Crane, R.L., S.A. Markel, and C. Fefferman, "SQP Algorithm Convergence in an Automated Environment", SIAM 94 presentation.

Apr-June'94

1. Crane, R.L., S.A. Markel, C. Fefferman and J. Pearson, "Convergence to Local Minima in Neural Network Training," submitted to NIPS, 1994.
2. Kung, S.Y., K.I. Diamantaras, and J.S. Taur, "Adaptive Principal Component EXtraction (APEX) and Applications", in press, IEEE Transactions on Signal Processing, May, 1994. (not attached)
3. Fefferman, C. and S.A. Markel, "Recovering a Feed-Forward Net From Its Output," NIPS, pg. 335-342, 1994

Jan-Mar'94

1. Diamantaras, K. and S.Y. Kung, "Multi-layer Neural Networks for Reduced-Rank Approximation", in press, IEEE Transactions on Neural Networks, 1994. (not attached).

Oct-Dec'93

1. Markel, S.A. and C. Fefferman, "Recovering a Feed-Forward Net From Its Output," NIPS Talk, 1993.
2. Fefferman, C. and S.A. Markel, "Recovering a Feed-Forward Net From Its Output," to appear in the proceedings of the 1993 NIPS Conference, "Advances in Neural Information Processing Systems 6", Morgan Kaufmann (1994).

Oct'92-Sep'93 (annual)

1. Markel, S.A., R.L. Crane and C. Fefferman, "Creating Good Random Starts for Neural Network Training", SIAM Talk, 1993.
2. Markel, S.A., R.L. Crane and C. Fefferman, "Counting Local Minima in Neural Network Training". SIAM Talk, 1993.
3. Diamantaras, K. and S.Y. Kung, "Principal Cross-correlation Component Neural Networks", in press, IEEE Transactions on Neural Networks, 1993. (not attached)

4. Kamm, C., G. Kuhn, B. Yoon, R. Chellappa, and S.Y. Kung (Co-Editors), "Neural Networks for Signal Processing, III," Publisher: IEEE Press, 1993. (not attached)
5. Diamantaras, K. and S.Y. Kung, "Compressing moving pictures using the APEX neural principal component extractor", Proc. IEEE Workshop on Neural Networks for Signal Processing, Baltimore, 1993.
6. Taur, J.S. and S.Y. Kung, "Fuzzy-decision neural networks and applications to data fusion", Proc. IEEE Workshop on Neural Networks for Signal Processing, Baltimore, 1993.
7. Sverdlow, R., "A Neural Net Application to Signal Identification", Proc. IEEE Workshop on Neural Networks for Signal Processing, Baltimore, 1993.

Apr-June'93

1. Markel, S.A. and R. L. Crane, "A New Algorithm for Efficient Hessian Calculation", submitted to NIPS 1993.
2. Fefferman, C., "Recovering a Feed-Forward Net from its Output", submitted to NIPS 1993.
3. De Vries, B., "Gradient-Based Adaptation of Network Structure", submitted to NIPS 1993.
4. Kung, S.Y. and J.S. Taur, "Decision-based Hierarchical Neural Networks with signal/Image Classification Applications", in press, IEEE Transactions on Neural Networks, 1993. (not attached)
5. Taur, J.S. and S.Y. Kung, "Fuzzy-Decision Neural Networks", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, April, 1993.

Jan-Mar'93

1. De Vries, B., "Gradient-Based Adaptation of Network Structure", Sarnoff Report, March, 1993.
2. De Vries, B., "Time-Varying Neural Networks for Large Tasks", submitted to ICANN 1993.
3. Taur, J.S. and S.Y. Kung, "Prediction-Based Networks with {ECG} Application", Proc. IEEE International Conference on Neural Networks, San Francisco, March 1993.
4. Kung, S.Y. and W. H. Chou, "Mapping neural networks onto VLSI array processors", Chapter 1 of "Parallel digital implementation of neural networks," K. W. Przytula and V.K. Prasanna, Prentice Hall, Englewood Cliffs, New Jersey, 1993. (not attached)
5. Kung, S.Y. "Digital Neural Networks", by Prentice-Hall, Inc., Englewood Cliffs, New Jersey, March 1993. (not attached)

Oct-Dec'92

1. De Vries, B., "Short Term Memory Structures for Dynamic Neural Networks", to appear in "Artificial Neural Networks with Applications in Speech and Vision", Richard J. Mammone, ed..
2. Kung, S.Y. and J.S. Taur, "Prediction-based Networks for Temporal Signal Classifications", to appear in "Artificial Neural Networks with Applications in Speech and Vision", Richard J. Mammone, ed..
3. Pearson, J.C. and Herb Taylor, "Neural Networks for Signal Processing", GOMAC 1992.
4. Fefferman, C., "Reconstructing a Neural Net from its Output", submitted to Revista Matematica Iberoamericana in January.